UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

SEDE VICTOR LEVI SASSO

FOLLETO DE INTRODUCCIÓN A LA TEORÍA COMPUTACIONAL

INCLUYE PRUEBAS SUMATIVAS Y PRESENTACIONES DEL CONTENIDO

DR. CARLOS A. ROVETTO

JUNIO 2021

Contenido

Índice de figuras

Índice de Tablas

Introducción

Capítulo I: Conceptos de lógica y razonamiento

En el Capítulo I de este documento se describirán conceptos básicos de lógica y razonamiento. Se presentará una introducción al estudio de la lógica como ciencia formal, brindando una definición a este concepto y una descripción acerca de su evolución a lo largo del tiempo. Se mencionarán otras disciplinas en las que se aplica la lógica, de modo que se pueda evidenciar la importancia del estudio de esta disciplina.

Posteriormente, se especificarán algunos de los diferentes tipos de lógica y se describirán detalladamente los dos tipos de lógica más importantes en el estudio de la teoría computacional: la *lógica proposicional* y la *lógica de predicado*. Ambos tipos de lógica son fundamentales para todo tipo de razonamiento lógico, por lo tanto, con el fin de lograr una adecuada comprensión de estos conceptos, se describirán los elementos y símbolos que forman parte de ambos tipos de lógica.

Finalmente, se estudiará la teoría de conjuntos. La teoría de conjuntos es uno de los temas elementales en el estudio de las matemáticas discretas, ya que casi todo objeto matemático que puede ser construido, se hace a partir de la serie de conjuntos (Epp, 2011). Debido a las numerosas aplicaciones de la teoría de conjuntos dentro del área de las ciencias computacionales, es de gran importancia comprender este tema. Se describirán los diversos tipos de conjuntos, las operaciones que pueden realizarse sobre estos y los tipos de relaciones que pueden existir entre conjuntos. Posteriormente, se describirá el concepto de diagrama de Venn y cómo estos facilitan la comprensión de la teoría de conjuntos y el análisis de problemas de lógica. Por último, se presentará un listado de las propiedades que pueden aplicarse a los problemas relacionados con conjuntos

Definición y conceptos de lógica

La lógica es una de las ramas teóricas de la filosofía, cuyo vocablo de origen griego significa “amor a la sabiduría” y se encarga de explicar y entender la naturaleza, las causas o los principios de la realidad, del conocimiento o de los valores, a través del razonamiento.

En la Figura 1 se muestran las ramas teóricas y prácticas de la filosofía:

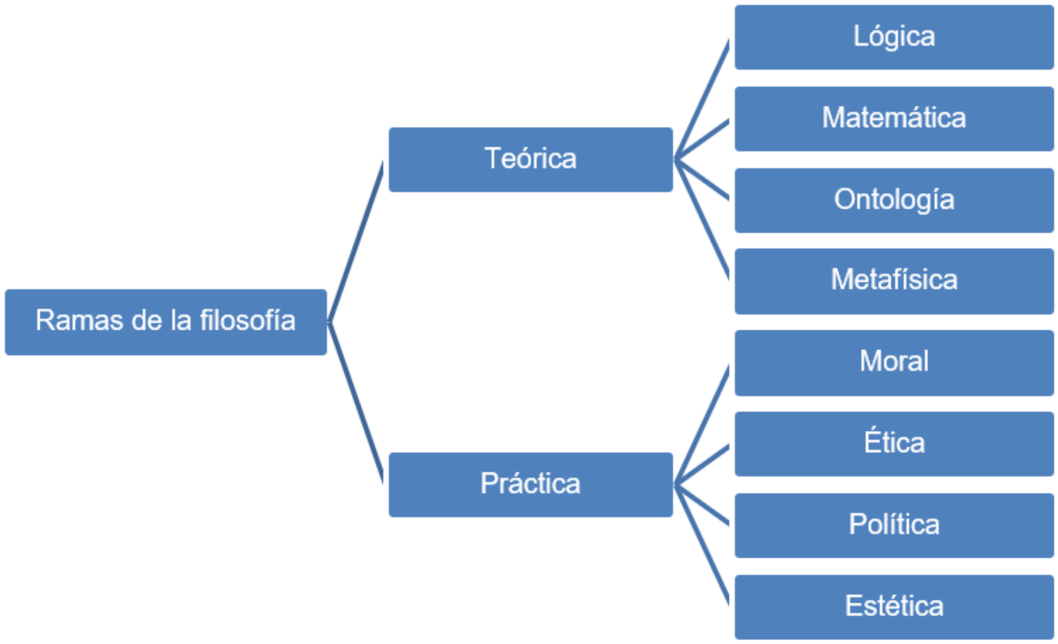


Figura 1. Ramas de la filosofía. Elaboración propia.

La **lógica** se puede definir como la disciplina que estudia el razonamiento a través de reglas y técnicas, que hacen posible determinar si un enunciado es cierto o falso (Jiménez Murillo, 2009). La lógica tiene como objetivo explicar un método de análisis para ordenar ideas, pensar correctamente, establecer conclusiones correctas y válidas y evitar el error en el razonamiento.

La lógica se subdivide en dos ramas principales: la lógica informal y la lógica formal. La lógica informal estudia el razonamiento, la inferencia y la argumentación del lenguaje natural, mientras que la lógica formal es el estudio abstracto de las proposiciones, argumentos y afirmaciones, a través de un conjunto de reglas.

El origen de la lógica se fundamenta en los estudios realizados por *Aristóteles*, los cuales posteriormente los griegos utilizaron para demostrar las leyes matemáticas de manera formal. Durante el siglo XIX, el filósofo y matemático alemán *Gottfried Leibniz* (Figura 2) propuso el uso de símbolos para representar y facilitar el proceso de razonamiento. La propuesta de Leibniz fue ampliada posteriormente por los matemáticos ingleses *George*

Boole (Figura 3) y *Augusto De Morgan*, quienes realizaron importantes aportes a la lógica matemática.

Figura 2. Gottfried Leibniz. Fuente: Encyclopedia Britannica.

Figura 3. George Boole. Fuente: Encyclopedia Britannica.

También destacan los trabajos de la Teoría de Conjuntos, realizados por los matemáticos alemanes *Gottlob Frege* y *Georg Cantor* (Figura 4). Entre finales del siglo XIX e inicios del siglo XX, los matemáticos ingleses *Bertrand Russell* (Figura 5) y *Alfred Whitehead* en sus estudios afirmaron que todas las leyes matemáticas pueden ser representadas a través proposiciones lógicas verdaderas, siendo esta teoría aún aceptada en la actualidad.

Figura 4. Georg Cantor. Fuente: Encyclopedia Britannica.

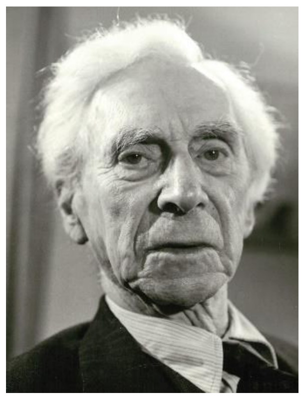


Figura 5. Bertrand Russell. Fuente: Encyclopedia Britannica.

Como describe (Jiménez Murillo, 2009), la lógica tiene numerosas aplicaciones en diversas disciplinas como la filosofía, las matemáticas, la física y las ciencias de la computación, entre otras. Dentro de este último conjunto de disciplinas, la lógica se aplica en el desarrollo de programas, en el estudio de los lenguajes formales, en la teoría de las bases de datos relacionales y además, son la base de los sistemas basados en tomas de decisiones, dentro del estudio de la Inteligencia Artificial. Adicional a esto, la lógica se aplica en nuestra vida cotidiana.

Parte del estudio de la lógica es establecer la validez de un argumento. Como define (Epp, 2011), un argumento es una secuencia de proposiciones (premisas) cuyo fin es demostrar que una afirmación (conclusión) es verdadera. Si las premisas de un argumento son verdaderas, entonces la conclusión también es verdadera y por consiguiente el argumento es válido.

El contenido de un argumento es lo que permite distinguir la forma del argumento. Esto es lo que se denomina **forma lógica** del argumento. Considere los siguientes ejemplos de argumentos:

Ejemplo #1: Si todos los gatos son carnívoros y Coco es un gato, por lo tanto Coco es carnívoro.

- **Forma lógica:** Si *p* y *q*, por lo tanto *r*.

Ejemplo #2: Si x= 3 o x= -3, entonces x^2= 9. Por lo tanto, si $x^2 \neq 9$ entonces x ≠ 3 o x ≠-3.

- **Forma lógica:** Si *p* o *q*, entonces *r*. Por lo tanto, si *no r* entonces no p o *no q*.

La representación de la forma lógica de una proposición es similar a los ejemplos presentados, con la diferencia de que se utiliza un sistema formal. Un sistema formal o cálculo lógico es cualquier sistema bien definido de pensamiento abstracto basado en el modelo de la matemática. El cálculo lógico es la base para el estudio de la lógica proposicional y la lógica de predicado.

Tipos de lógica

Existen diversos tipos de lógica, como se muestra en la Figura 6:

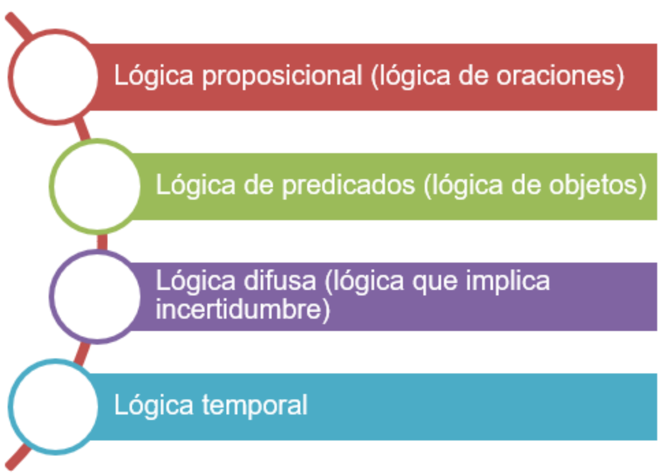


Figura 6. Algunos tipos de lógica. Elaboración propia.

Ambos sistemas formales, tanto la lógica de predicados como el lenguaje proposicional, consisten en las siguientes partes:

- **Sintaxis:** Alfabeto y reglas para la formación de fórmulas bien formuladas (*wffs*).
- **Semántica:** Preguntas que conectan modelos: cómo traducir un *wff* por ejemplo en lenguaje natural (interpretación) y cuáles son las condiciones exactas de su verdad (determinación de valores de verdad).
- **Prueba (proof):** Es el principio del cálculo, los axiomas y de las reglas de inferencia. Los teoremas se deducen de los axiomas por medio de reglas de inferencia. Los axiomas son verdades o premisas evidentes que no requieren una demostración. Así, deben ser lógicamente cierto.

Por otro lado, la lógica difusa estudia el razonamiento de argumentos imprecisos y que pueden tener valores aleatorios, mientras que la lógica temporal estudia el razonamiento de argumentos definidos en conceptos relacionados con el tiempo.

Lógica proposicional

La **lógica proposicional** es una lógica en el nivel de la sentencia, en el que la unidad más pequeña es una oración. En este tipo de lógica no interesan las oraciones individuales o el significado de estas, sino saber si las oraciones son verdaderas o falsas. También interesa saber si el hecho de que una oración sea verdadera o falsa depende de un conjunto de oraciones, por lo que de ser afirmativo, entonces es relevante saber cómo se obtiene. Estas oraciones se denominan proposiciones.

Como define (Jiménez Murillo, 2009)(Jiménez Murillo, 2009), una **proposición** es una oración, frase o expresión matemática que es verdadera o falsa, pero no ambas a la vez. Considerando esto, para que una oración se pueda considerar como una proposición **válida**, esta debe tener bien definida sus **valores de verdad**, es decir, que debe ser verdadera o falsa. Si una proposición es verdadera, entonces se dice que tiene un valor de verdad de “verdadero”; si una proposición es falsa, su valor de verdad es “falso”.

Considere los siguientes ejemplos de proposiciones:

a. “La hierba es verde”

b. “2 + 5 = 5”

c. “Los gatos son mamíferos”

d. “Rusia es el país más extenso del mundo”

Las proposiciones **a**, **c** y **d** tienen una valor de verdad de “verdadero”, mientras que la proposición **b** tiene un valor de verdad de “falso”.

Ahora, considere los siguientes ejemplos:

e. “Cierra la puerta”

f. “¿Hace calor afuera?”

g. “x = 2”

h. “x es mayor que 5”

Los enunciados **e**, **f**, **g** y **h** no son proposiciones. Las oraciones **e** y **f** no pueden tomar un valor de verdadero o falso, mientras que en el caso de las oraciones **g** y **h**, la *x* es una variable que representa un número. Para que estos dos enunciados puedan ser considerados proposiciones, se debe dar el valor específico de la *x* y así saber qué representa y decir si es verdadero o falso.

Las oraciones presentadas en el ejemplo anterior son proposiciones básicas y sencillas. Sin embargo, es posible construir proposiciones más grandes y complejas combinando proposiciones primitivas con **conectivas lógicas**, como explica (Conradie & Goranko, 2015). En este sentido, las proposiciones y las conectivas son los **elementos básicos** de la lógica proposicional. Las conectivas lógicas más comunes son las siguientes, mostradas en la Tabla 1:

Conectiva	Significado
¬	Not
∧	And
∨	Or
→	Si --- Implica
↔	Si y solo si

Tabla 1. Conectivas lógicas más comunes.

Lógica de predicados

La lógica proposicional no es lo suficientemente poderosa como para representar todo tipo de aserciones que se utilizan en la informática y las matemáticas, o para expresar ciertos tipos de relación entre proposiciones, tales como la equivalencia.

Por ejemplo, considere la siguiente aserción: “x es mayor que 1”. En esta aserción la *x* es una variable y por lo tanto, no es una proposición porque no se puede decir si es verdadera o falsa a menos que se conozca el valor de *x*.

Así, la lógica proposicional no puede ocuparse de tales oraciones. Sin embargo, tales aserciones aparecen con bastante frecuencia en matemáticas y se requiere realizar inferencias sobre esas afirmaciones.

También el patrón implicado en las equivalencias lógicas siguientes no puede ser capturado por la lógica proposicional. Como ejemplo de esto, observe las siguientes proposiciones:

- **a.** “No todos los pájaros vuelan” **es equivalente a** “Algunos pájaros vuelan”.
- **b.** “No todos los números enteros son pares” **es equivalente a** “Algunos números enteros no son iguales”.
- **c.** “No todos los autos son caros” **es equivalente a** “Algunos autos no son caros”.

Cada una de estas proposiciones se trata independientemente de las otras en lógica proposicional. Por ejemplo, considerando el ejemplo **a**, si **P** representa “No todos los pájaros vuelan” y **Q** representa “Algunos pájaros vuelan”, dentro de la lógica proposicional no existe ningún mecanismo para descubrir que **P** y **Q** son equivalentes. Si sólo se usa la lógica proposicional, para poder utilizar estas equivalencias en una inferencia, estas se deben listar de forma individual en lugar de tratar de cubrirlas de forma colectiva con una fórmula general e instanciarlas cuando sean necesarias.

Debido a esto, es necesaria una lógica más poderosa que permita lidiar con estos y otros problemas. La **lógica de predicado** es una de esas lógicas y aborda estas cuestiones, entre otras. De esta manera es posible presentar expresiones más complicadas del lenguaje natural y utilizarlas en la inferencia formal.

Como estándar se denota el lenguaje predicado puro con el símbolo **P** (en minúscula **ρ**). El alfabeto **P** consiste en los siguientes símbolos:

- **Conectivas:** ¬ , ∧, ∨, →, ↔
- **Punto y paréntesis**
- **Constantes individuales:** a, b, c,…
- **Variables:** x, y, z,…
- **Símbolos del predicado:** P, Q, R,…
- **Símbolos para la función:** f, g,…
- **Símbolo de identidad:** =
- **Cuantificadores:** ∀ y ∃

Las **constantes** se utilizan para denotar nombres de criaturas del mundo real o ideal. De forma más precisa, la interpretación relaciona constantes a algunos elementos de un conjunto A. Las **variables** también se refieren a los elementos de un conjunto, pero no los determina explícitamente.

En la lógica de predicados los términos suelen usarse en lugar de sustantivos y pronombres. Se combinan en oraciones por medio de predicados. Por ejemplo, en la frase "Pedro trabaja con Mario" los sustantivos son "Pedro" y "Mario", y el predicado es "trabaja con". Lo mismo es cierto si esta frase se traduce en lógica de predicado, excepto que "Pedro" y "Mario" ahora se llaman **términos**.

La lógica de predicado hace frente a las deficiencias de la lógica proposicional introduciendo dos nuevas elementos: predicados y cuantificadores.

Un **predicado** es una plantilla de frase verbal que describe una propiedad de objetos o una relación entre objetos representados por las variables.

Considere los siguientes ejemplos de oraciones:

a. "El automóvil que PEDRO está manejando es verde"

b. “Esta manzana es verde”

c. “La portada de este libro es verde”

Las tres oraciones presentadas provienen de la plantilla “es verde”, colocando una frase sustantiva/nominal apropiada delante de la misma. La frase “es verde” es un predicado y describe la propiedad de ser verde.

Generalmente, a los predicados se les da un nombre. Para el caso presentado anteriormente, para representar el predicado “es verde”, puede usarse “es_verde”, “Verde” o “V”, entre otros. Si se adopta “V” como el nombre del predicado “es verde”, las oraciones que afirman que un objeto es verde pueden representarse como “V(x)”, donde *x* representa un objeto arbitrario. “V(x)” se lee como “x es verde”.

A continuación se presenta otro ejemplo; para ello considere las siguientes oraciones:

a. “PEDRO le da el libro a JUAN”

b. “ANA le da una pizza a SARA”

c. “MARCOS le da una explicación a MARIO”

En este caso, estas oraciones se obtienen sustituyendo un objeto apropiado por las variables ***x***, ***y***, ***z*** en la oración “x da y a z”. Se observa que el predicado es la plantilla “…da…a…” y describe una relación entre tres objetos. Este predicado puede representarse, por ejemplo, como “Dar(x, y, z)” o “G(x, y, z)”.

En la lógica predicados, cada predicado recibe un nombre, seguido de la lista de los argumentos, como es el caso de la representación del predicado del ejemplo anterior. Muy a menudo sólo se utilizan letras únicas para nombres de predicados y términos. El orden de los argumentos es importante. Considerando el ejemplo anterior, las afirmaciones G(x, y, z) y G(x, z, y) tienen significados completamente diferentes.

Estas afirmaciones, en el lenguaje predicado **P**, se denominan fórmulas atómicas o átomos. Una fórmula atómica es un nombre de predicado seguido de una lista de argumentos o una identidad t1 = t2, donde t1 y t2 son términos (es decir, constantes individuales o variables). Son declaraciones y pueden ser combinadas por conectivas lógicas de la misma manera que las unidades en la lógica proposicional. Por ejemplo:

$$G(x, y, z) \rightarrow \neg G(x, z, y)$$

Como se observa en las fórmulas atómicas anteriores, se utilizan variables en los argumentos. Ocasionalmente, uno no quiere asociar los argumentos de una fórmula atómica con un elemento en particular y para evitarlo, se utilizan variables. Los nombres de las variables son frecuentemente escogidos desde el final del alfabeto: x, y, z, etc., con o sin subíndices. También se les puede dar nombres a las expresiones (meta-variables).

Considere la siguiente afirmación: "Si x es un gato entonces x tiene cola". Esto puede expresarse formalmente como: G(x):= "x es un gato", C(x):= "x tiene cola". La formalización de la afirmación completa sería la siguiente:

$$A(x) : = G(x) \rightarrow C(x)$$

Antes de definir el siguiente elemento de la lógica de predicados, conviene definir el concepto de *universo del discurso*. El **universo del discurso**, también denominado **universo**, es el dominio de las variables (individuales); es la colección de todas las personas, ideas, símbolos, estructuras de datos, etc., que afectan al argumento lógico bajo consideración. A menudo, el universo a menudo se deja implícito en la práctica, **pero debe ser obvio desde el contexto**.

Los **cuantificadores** son frases especiales que a menudo se usan para cuantificar objetos en el discurso (Conradie & Goranko, 2015). En otras palabras, para indicar si una sentencia es siempre verdadera, si es a veces verdadera, o nunca es verdadera. En este sentido, los cuantificadores se utilizan para corresponder palabras tales como "Todo", "algunos", "nunca", y expresiones relacionadas.

Un predicado con variables no es una proposición, sin embargo, para convertirla en una proposición se puede aplicar alguna de las siguientes operaciones a la variable: *a)* asignar un valor a la variable o *b)* cuantificar la variable utilizando un cuantificador.

Por ejemplo, la sentencia “$x > 3$” tiene como variable a *x* sobre el universo de números reales y es una sentencia que no es verdadera ni falsa, ya que no se sabe el valor de *x* y de esto dependerá si la sentencia es verdadera o falsa. Si a *x* se le asigna el valor de 5, la sentencia “$x > 3$” se convierte en “$5 > 3$”, lo cual la convierte en una sentencia verdadera y por consiguiente, en una proposición.

En general, se realiza una cuantificación en fórmulas de lógica de predicados (denominadas *wff*), tales como “$x > 3$” o “P(x)”, utilizando cuantificadores de variables. Se pueden distinguir dos tipos de cuantificadores:

- **Cuantificador universal:** Su símbolo es **∀** y se denota a través de la expresión **∀x P(x)**, la cual denota la cuantificación universal de la fórmula atómica **P(x)**. Se lee “para todo *x*, *P(x)* tiene”, “para cada *x*, *P(x)* tiene” o “para cada *x*, *P(x)* mantiene”. En dicha expresión, la *x* significa *todos los objetos x en el universo* y como está seguido por *P(x)* entonces significa que *P(x)* es verdadero para cada objeto *x* en el universo.

  Por ejemplo, “Todos los gatos son mamíferos” puede convertirse en la forma proposicional ∀x P(x), donde el predicado *P(x)* denota “son mamíferos” y el universo del discurso sólo está conformado por gatos.

  Si todos los elementos del universo del discurso pueden ser enumerados, entonces la cuantificación universal ∀x P(x), es equivalente a la conjunción: $\forall x\ P(x_1) \wedge P(x_2) \wedge P(x_3) \ldots \wedge P(x_n)$.

  Por ejemplo, en el ejemplo anterior de ∀x P(*x*), si supiéramos que sólo hay tres gatos en el universo del discurso (g_1, g_2 y g_3), también podríamos traducir la sentencia como: $P(g_1) \wedge P(g_2) \wedge P(g_3)$.

- **Cuantificador existencial:** Su símbolo es **∃** y se denota a través de la expresión **∃x P(*x*)**, la cual denota la cuantificación existencial de la fórmula atómica **P(x)**. Se lee “existe una x tal que *P(x)*” o “hay al menos un *x* tal que *P(x)*”. En dicha expresión, la *x* significa *al menos un objeto x en el universo* y como está seguido por *P(x)* entonces significa que *P(x)* es verdadero para al menos un objeto *x* en el universo.

Por ejemplo, “Alguien compró un boleto” puede convertirse en la forma proposicional ∃x P(*x*), donde el predicado *P(x)* denota “x compró un boleto” y el universo del discurso contiene (pero no se limita) a todos los humanos.

Si todos los elementos del universo del discurso pueden ser enumerados, entonces la cuantificación universal ∃x P(*x*), es equivalente a la conjunción: ∃x $P(x_1) \vee P(x_2) \vee P(x_3) \ldots \vee P(x_n)$.

Por ejemplo, en el ejemplo anterior de ∃*x* P(*x*), si supiéramos que sólo hay tres entidades en el universo del discurso (yo, él y ella), también podríamos traducir la sentencia como: P(yo) ∨ P(él) ∨ P(ella).

Las fórmulas cuantificadas se deben leer **de izquierda a derecha** y los cuantificadores pueden leerse como:

- **∀x** puede leerse como “para cada/todo objeto *x* en el universo se sustenta/sostiene/dice”.
- **∃*x*** puede leerse como “existe un objeto *x* en el universo que satisface el siguiente” o “para algún objeto *x* en el universo lo siguiente se cumple”.

Considere el siguiente ejemplo: Sea el universo el conjunto de gatos y sea G(x,y) el predicado “*x* corre más rápido que *y*”. Entonces:

- **∀x ∀y G(x, y)** puede leerse como “Para cada gato *x* se cumple lo siguiente: *x* es más rápido que cada (cualquier) gato *y*”. Una forma más sencilla sería “Cada gato es más rápido que cada gato”.
- **∀x ∃y G(x, y)** puede leerse como “Para cada gato *x* se cumple lo siguiente: para algún gato *y*, *x* es más rápido que *y*”. Una forma más sencilla sería “Cada gato es más rápido que un gato”.
- **∃x ∀y G(x, y)** puede leerse como “Existe un gato *x* que satisface lo siguiente: (o tal que) para cada gato *y*, *x* es más rápido que *y*”. Una forma más sencilla sería “Hay un gato que es más rápido que cada gato”.

- **∃x ∃y G(x, y)** puede leerse como “Para un gato *x* existe un gato *y* tal que *x* es más rápido que *y*”, lo cual de forma más sencilla sería “Un gato es más rápido que un gato”.

De igual manera, es importante tener en cuenta el **orden de aplicación de los cuantificadores**. Las posiciones del mismo tipo de cuantificadores pueden conmutarse sin afectar el valor de verdad, siempre que no haya cuantificadores del otro tipo entre los intercambiables. Por ejemplo, ∃x ∃y ∃z, P(x, y, z) es equivalente a ∃y ∃x ∃z P(x, y, z), … etc. Sin embargo, ∀x ∃y F(x, y) no es equivalente a ∃y ∀x F(x, y), ya que los tipos diferentes de cuantificadores no se pueden intercambiar.

- **Reglas para construir fórmulas bien formuladas (wff)**

No todas las cadenas pueden representar proposiciones de la lógica de predicado. Para que los símbolos produzcan una proposición y sean bien interpretados deben seguir las reglas dadas abajo, y se llaman **wffs** (fórmulas bien formadas) de la lógica del predicado del primer orden. Un nombre de predicado seguido por una lista de variables tales como *P(x, y)*, donde P es un nombre de predicado y *x, y* son variables, se denomina fórmula atómica. Las *wffs* se construyen utilizando las siguientes reglas:

- Las *wffs* son verdadera y falsa.
- Cada constante proposicional (es decir, proposición específica), y cada variable proposicional (es decir, una variable que representa proposiciones) son *wffs*.
- Cada fórmula atómica (es decir, un predicado específico con variables) es un *wffs*.
- Si A, B y C son *wffs*, también lo son ¬ A, (A ∧ B), (A ∨ B), (A ⟶ B) y (A ⟷ B).

Si *x* es una variable (que representa objetos del universo del discurso), y A es un *wff*, entonces también lo son **∀x** A y **∃x** A.

- **Cálculo de predicado**

A continuación se presentan ejemplos de cálculo de predicado:

- **Ejemplo #1:** Pedro habla con todo el mundo.
  Respuesta: ∀x habla(Pedro, x)

- **Ejemplo #2:** Todo el mundo se ama a sí mismo.
  Respuesta: $\forall x$ ama(x, x)

- **Ejemplo #3:** Llovió el viernes.
  Respuesta: Clima(viernes, lluvia)

- **Ejemplo #4:** Llovió todos los días.
  Respuesta: Tiempo(x, lluvia) $\wedge$ día(x)

- **Ejemplo #5:** Si no llueve mañana, Pedro irá al parque.
  Respuesta: $\neg$Tiempo(lluvia, mañana) $\longrightarrow$ ir(Pedro, parque)

- **Ejemplo #6:** Todos los jugadores de baloncesto son altos.
  Respuesta: $\forall x \forall y$ (jugadores(x) $\wedge$ juegan(x, y) $\wedge$ baloncesto(y)) $\longrightarrow$ altos(x)

- **Ejemplo #7:** Algunas personas comen mariscos.
  Respuesta: $\exists x$ (personas(x) $\wedge$ comen(x, mariscos))

- **Ejemplo #8:** A nadie le gusta los impuestos.
  Respuesta: $\neg\exists x$ gusta(x, impuestos)

Teoría de Conjuntos

La **teoría de conjuntos** fue propuesta inicialmente por el matemático alemán Georg Cantor durante el final del siglo XIX, tras descubrir la potencial utilidad que ofrecía el estudio de los conjuntos en general en lugar de estudiar las propiedades de los elementos que componen a esos conjuntos (Jiménez Murillo, 2009).

Aunque en un principio otros matemáticos de la época se opusieron a las ideas de Cantor, en la actualidad la teoría de conjuntos es la base para el estudio de diversas áreas de las matemáticas y principalmente, de las ciencias computacionales. Dentro de esta última área, como describe (Conradie & Goranko, 2015), la teoría de conjuntos es la base de temas como el álgebra booleana, los autómatas, las bases de datos, las redes, entre otros.

Un **conjunto** es una colección de objetos considerada como un todo. Esta colección *está bien definida*, lo cual significa que para cada objeto que pueda formar parte de un conjunto, existe una forma de determinar si ese objeto forma o no parte del conjunto.

Los objetos de un conjunto son llamados **elementos** o miembros del conjunto. Los elementos de un conjunto pueden ser cualquier cosa: números, personas, letras, otros conjuntos, etc. Todos los elementos que pertenecen a un conjunto son del mismo tipo y esos elementos son distintos, unos de otros, sin que se repitan dentro de un conjunto. Además, el orden de los elementos no es importante.

Ejemplo: Considere el siguiente conjunto que contiene como elementos a las letras de la palabra "verde". Se pueden eliminar los elementos repetidos y se pueden listar en cualquier orden y se estaría representando el mismo conjunto, como se observa a continuación:

$$A = \{v, e, r, d, e\}$$

$$A = \{v, e, r, d\}$$

$$= \{d, e, r, v\}$$

Los conjuntos se denotan por letras mayúsculas: A, B, C, etc., y los elementos se denotan por letras minúsculas, números o combinación de ambos (Conradie & Goranko, 2015). Un conjunto se expresa colocando los elementos dentro de llaves **{ }** y separados por comas. Un conjunto se puede especificar de dos maneras:

a. Por extensión: Enumerando sus elementos.

Ejemplo #1: $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Ejemplo #2: $B=\{a, e, i, o, u\}$

b. **Por comprensión:** Indicando alguna propiedad que verifique todos y cada uno de los elementos del conjunto, y sólo a ellos.

Ejemplo #1: *A= {vocales del alfabeto}*

Ejemplo #2: *C= {x | x es un entero impar, x>10}*

Nota: El símbolo "**|**" significa "tal que" y la coma "**,**" significa "y".

También existen algunos conjuntos especiales, los cuales se definen a continuación:

- **Conjunto universal:** También denominado **universo**, es aquel que está formado por todos los elementos que se están considerando. Se representa por U, A o Ω. Como explica (Lipschutz & Lipson, 2009), todos los conjuntos en cualquier aplicación de la teoría de conjuntos pertenecen a un gran conjunto universal.

- **Conjunto vacío:** Es aquel que no contiene elementos y se denota por el símbolo ∅ o como {}. El conjunto vacío también se considera

- **Conjunto unitario:** Es aquel que está formado por un único elemento.

Se puede expresar la relación de pertenencia del elemento de un conjunto. Para denotar que un elemento *pertenece* a un conjunto se utiliza el símbolo ∈, mientras que para denotar que un elemento *no pertenece* a un conjunto se utiliza el símbolo ∉. Por ejemplo

Ejemplo: Considere el siguiente conjunto:

$A = \{a, e, i, o, u\}$ → Se tiene que:

- $a \in A$, es decir, a pertenece al conjunto A.
- $t \notin A$, es decir, t no pertenece al conjunto A.

La **cardinalidad de un conjunto** se refiere a la cantidad de elementos que contiene un conjunto. La cardinalidad de un conjunto A se denota como $|A|$, $card(A)$ o $n(A)$.

Ejemplo #1: Indique la cardinalidad de $B = \{1, 3, 5, 7, 9, 13\}$.

- **Solución:** $|B| = 6$

Ejemplo #2: Indique la cardinalidad de $A = \{x \mid x \; es \; una \; vocal\}$.

- **Solución:** $|A| = 5$

Ejemplo #3: Indique la cardinalidad de $C = \{13, 14, \ 15, \dots, 47, 48, 49\}$.

- **Solución:** $|C| = 37$. La cardinalidad de C es igual a la resta $49 - 13 = 36$, más el 13 incluido.

Un conjunto puede contener una cantidad infinita de elementos.

También es posible establecer la **igualdad de conjuntos**. Si se tienen dos conjuntos A y B, y cada elemento del conjunto A es también elemento del conjunto B, entonces ambos conjuntos *son iguales*. Esto se cumple si ambos conjuntos tienen los mismos elementos, es decir si $x \in A$ y $x \in B$, o si ambos son vacíos.

Ejemplo: Considere los siguientes conjuntos e indique si $A = \ B$ y $C = \ D$

$A = \{0, 3, 6, 9\}$

$B = \{0, 9, 6, 3, 6, 6, 9, 0\}$

$\longrightarrow$ $A = B$ porque tienen los mismos elementos. Recordemos que se pueden eliminar los elementos repetidos y que el orden de estos no importa, el conjunto B quedaría de la siguiente manera. También se cumple que $B = A$

$C = \{a, e, i, o, u\}$

$D = \{a, e, o\}$

$\longrightarrow$ $C \neq D$ porque $i \notin D$ y $u \notin D$, aunque los elementos del conjunto D están en el conjunto C.

Otro concepto importante dentro de la teoría de conjuntos es la de **subconjunto**. Cuando dos conjuntos A y B son iguales, se cumple que $A \subseteq B$, es decir que *A es un subconjunto de B*, *A es una parte de B* o *A está incluido en B*. Si se quiere representar que *A no es subconjunto de B* se haría de la siguiente forma $A \nsubseteq B$.

Ejemplo: Considere los conjuntos del ejemplo anterior e indique si $A \subseteq B$, $B \subseteq A$, $C \subseteq D$ y $D \subseteq C$.

$A = \{0, 3, 6, 9\}$

$B = \{0, 9, 6, 3, 6, 6, 9, 0\}$

$\longrightarrow$ $A \subseteq B$ porque cada elemento de A es también elemento de B. De igual manera, $B \subseteq A$.

$C = \{a, e, i, o, u\}$

$D = \{a, e, o\}$

$\longrightarrow$ $C \nsubseteq D$ porque $i \notin D$ y $u \notin D$. Sin embargo, $D \subseteq C$ porque los elementos de D están contenido en C.

También existe el **subconjunto propio**. Suponga que A y B son conjuntos tales que todo elemento de A es también elemento de B y se cumple que $A \neq B$ y $A \subseteq B$, entonces se dice que *A es un subconjunto propio de B* o *A está contenido estrictamente en B*. Esto se representa como $A \subset B$, mientras que si se quiere representar que A no es subconjunto de B se haría de la siguiente forma $A \not\subset B$.

Ejemplo: Considere los siguientes conjuntos:

$C = \{a, e, i, o, u\}$

$D = \{a, e, o\}$

$\longrightarrow$ Se cumple que $C \neq D$, y que $D \subseteq C$ porque $a \in D$, $e \in D$ y $o \in D$. También se observa que $C \nsubseteq D$. Con esto, se concluye que $C \subset D$.

$A = \{1, 2, 3, 4, 5\}$

$B = \{2, 4\}$

$\longrightarrow$ Se cumple que $A \neq B$ y que $B \subseteq A$, porque $2 \in D$ y $4 \in D$. Por lo tanto, $B \subset A$.

Como describe (Conradie & Goranko, 2015), la definición de subconjunto permite obtener que:

- **a.** Todo conjunto es subconjunto de sí mismo, es decir, $A \subseteq A$.
- **b.** El conjunto vacío es subconjunto de todos los conjuntos y de él mismo. Es decir, $\emptyset \subseteq A$, $\emptyset \subseteq U$ y $\emptyset \subseteq \emptyset$.

c. Todos los conjuntos son subconjuntos del conjunto universo. Es decir, $A \subseteq U$, $\emptyset \subseteq U$ y $U \subseteq U$.

El **conjunto potencia** de un conjunto A son todos los subconjuntos de A. Se representa por $P(A)$ y el número de subconjuntos de este se denota por $|P(A)| = 2^n$, donde n es número de elementos del conjunto A.

Ejemplo #1: Obtener el conjunto potencia del conjunto $A = \{1, 3, 5\}$ e indicar el número de subconjuntos de $P(A)$.

- **Solución:** La cantidad de elementos del conjunto A es $n = 3$, por lo que:

$$|P(A)| = 2^3 = 8$$

Entonces el conjunto potencia de A es:

$$P(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}, \{1,5\}, \{3, 5\}, \{1, 3, 5\}\}$$

Ejemplo #2: Obtener el conjunto potencia del conjunto $C = \{a\}$ e indicar el número de subconjuntos de $P(A)$.

- **Solución:** La cantidad de elementos del conjunto A es $n = 1$, por lo que:

$$|P(A)| = 2^1 = 2$$

Entonces el conjunto potencia de A es: $P(A) = \{\emptyset, \{a\}\}$

Ejemplo #3: Obtener el conjunto potencia del conjunto $C = \{a, \{b, c\}, d\}$ e indicar el número de subconjuntos de $P(A)$.

- **Solución:** La cantidad de elementos del conjunto A es $n = 3$, por lo que:

$$|P(A)| = 2^3 = 8$$

Entonces el conjunto potencia de A es:

$$P(A) = \{\emptyset, \{a\}, \{\{b, c\}\}, \{d\}, \{a, \{b, c\}\}, \{a, d\}, \{\{b, c\}, d\}, \{a, \{b, c\}, d\}\}$$

Operaciones

Existen varias formas de obtener nuevos conjuntos a partir de otros existentes. Las operaciones básicas son las siguientes:

a. **Unión:** Si se tienen dos conjuntos A y B, la unión es el conjunto de todos los elementos de A o de B. Se denota por $A \cup B$. Simbólicamente, se representa como $A \cup B = \{x \mid x \in A \vee x \in B\}$.

   Ejemplo #1: A partir de los conjuntos $A = \{1, 2, 3, 4\}$ y $B = \{-1, -2, 3, 4\}$, obtenga $A \cup B$.

   - **Solución:** $A \cup B = \{-1, 1, -2, 2, 3, 4\}$

   Ejemplo #2: A partir de los conjuntos $C = \{rojo, amarillo, azul\}$ y $D = \{morado, naranja, verde\}$, obtenga $C \cup D$.

   - **Solución:** $C \cup D = \{rojo, amarillo, azul, morado, naranja, verde\}$

b. **Intersección:** Si se tienen dos conjuntos A y B, la intersección es el conjunto de los elementos que pertenecen tanto a A como a B. Se denota por $A \cap B$. Simbólicamente, se representa como $A \cap B = \{x \mid x \in A \wedge x \in B\}$.

   Ejemplo #1: A partir de los conjuntos $A = \{1, 2, 3, 4\}$ y $B = \{-1, -2, 3, 4\}$, obtenga $A \cap B$.

   - **Solución:** $A \cap B = \{3, 4\}$

   Ejemplo #2: A partir de los conjuntos $C = \{a, b, c, d, e\}$ y $D = \{f, g, h, e, i, z\}$, obtenga $C \cap D$.

   - **Solución:** $A \cap B = \{e\}$

c. **Diferencia:** Si se tienen dos conjuntos A y B, la diferencia de A menos B es el conjunto de los elementos que pertenecen a A pero no pertenecen a B. Se denota por $A - B$. Simbólicamente, se representa como $A - B = \{x \mid x \in A \wedge x \notin B\}$.

   Ejemplo #1: A partir de los conjuntos $A = \{1, 2, 3, 4\}$ y $B = \{-1, -2, 3, 4\}$, obtenga $A - B$.

- **Solución:** $A - B = \{1, 2\}$

Ejemplo #2: A partir de los conjuntos $C = \{a, b, c, d, e\}$ y $D = \{f, g, h, e, i, z\}$, obtenga $D - C$.

- **Solución:** $D - C = \{f, g, h, i, z\}$

d. **Complemento:** Si se tiene un conjunto A, el complemento de A es el conjunto de todos los elementos que pertenecen al universo, pero que no pertenecen a A. Se denota por A^C, A' o $\bar{A}$. Simbólicamente, se representa como $A^C = \{x \mid x \in U \wedge x \notin A\}$.

Ejemplo #1: Suponga que $U = \{-5, -3, 0, 1, -7, 2, 4, 6, 8\}$ y que se tienen los conjuntos $A = \{0, 2, 4, 6, 8\}$ y $B = \{-5, -3, 1, -7\}$. Obtenga A^C.

- **Solución:** $A^C = \{-5, -3, 1, -7\}$

Ejemplo #2: Suponga que $U = \{-1, 0, -3, 2, -5, 6, 4, 8, 3\}$ y que se tienen los conjuntos $A = \{2, 4, 6, 8\}$ y $B = \{0, 3\}$. Obtenga B^C.

- **Solución:** $B^C = \{-1, -3, -5, 2, 4, 6, 8\}$

Ejemplo #3: Suponga que $U = \{a, b, c, \dots, x, y, z\}$ y que se tienen los conjuntos $C = \{a, e, i, o, u\}$ y $D = \{consonantes\ del\ alfabeto\}$. Obtenga D^C.

- **Solución:** $D^C = \{a, e, i, o, u\}$

Productos cartesianos y relaciones

Una **relación** se define como la correspondencia por medio de ciertas características, entre dos elementos de dos conjuntos. Es decir, si se tienen dos conjuntos A y B, el primer elemento a del conjunto A está relacionado con el segundo elemento b del conjunto B. Se denota como aRb.

Los tipos de relaciones básicas de conjuntos que se pueden identificar son los siguientes, descritos por (Conradie & Goranko, 2015):

a. **Relación reflexiva:** Todo elemento de un conjunto A está relacionado consigo mismo, de modo que se cumple que $(a, a) \in R$.

b. **Relación simétrica:** Si se tienen dos conjuntos A y B, se tiene una relación simétrica si se cumple que $(a,b) \in R$ y $(b,a) \in R$. También puede cumplirse que $(a,b) \notin R$ y $(b,a) \notin R$.

c. **Relación transitiva:** Si se tienen tres conjuntos A , B y C, se tiene una relación transitiva si se cumple que $(a,b) \in R$ y $(b,c) \in R$ por lo que $(a,c) \in R$.

Otro tipo de relación es el producto cartesiano. El **producto cartesiano** de dos conjuntos A y B es la combinación de todos los elementos de A con todos los elementos de B (Conradie & Goranko, 2015). Es un conjunto de pares ordenados de modo que $A \times B = \{(a,b) \mid a \in A, b \in B\}$. Considere los conjuntos A y B con sus elementos, el producto cartesiano sería como se muestra a continuación:

$$\begin{matrix} A = \{a_1, a_2\} \\ B = \{b_1, b_2\} \end{matrix} \longrightarrow A \times B = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2)\}$$

Esto puede representarse gráficamente de la siguiente manera, como se observa en la Figura 7:

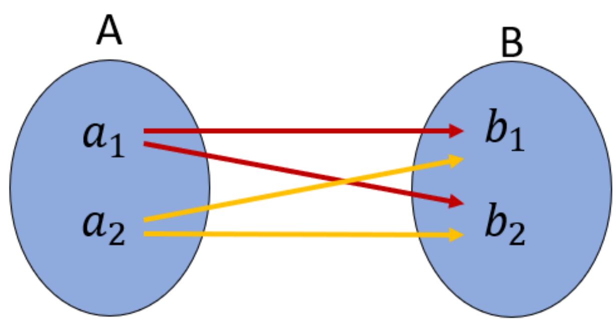


Figura 7. Producto cartesiano de dos conjuntos. Elaboración propia.

El producto cartesiano de dos conjuntos no es conmutativo, es decir, que $A \times B \neq B \times A$. Además, se puede obtener la cantidad de pares ordenados que tendrá el producto cartesiano de dos conjuntos A y B, multiplicando la cardinalidad de ambos conjuntos. En tal caso, si $|A| = x$ y $|B| = y$, entonces la cantidad de pares ordenados de $A \times B$ será igual a $x \cdot y$.

Ejemplo #1: Obtenga el producto cartesiano de los conjuntos $A = \{2, 4, 6\}$ y $B = \{1,3\}$.

- **Solución:** $A \times B = \{(2,1),(2,3),(4,1),(4,3),(6,1),(6,3)\}$

Ejemplo #2: Obtenga el producto cartesiano de los conjuntos $A = \{a, b, c, d\}$ y $B = \{1,3,5\}$.

- **Solución:**

$$A \times B = \{(a,1),(a,3),(a,5),(b,1),(b,3),(b,5),(c,1),(c,3),(c,5),(d,1),(d,3),(d,5)\}$$

Diagramas de Venn

Los diagramas de Venn fueron desarrollados por el matemático y filósofo inglés John Venn (1834 - 1923).

Un **diagrama de Venn** es un gráfico que representa la relación entre los elementos de un conjunto, con regiones encerradas dentro de un plano (Lipschutz & Lipson, 2009). El universo se representa con un rectángulo y los conjuntos con regiones circulares. Se sombrea el área que se desea ilustrar.

Como explica (Conradie & Goranko, 2015), la manera en la que se entrelazan los gráficos, representa la relación que existe entre los elementos de los conjuntos. Debido a esto, los diagramas de Venn facilitan la comprensión y resolución de problemas de lógica.

En la Figura 8 se muestra un ejemplo de un diagrama de Venn, en donde el rectángulo es el universo o conjunto universal y los círculos son conjuntos (A, B y C). Además, se puede afirmar que A, B y C son subconjuntos de U y que C es un subconjunto de B.

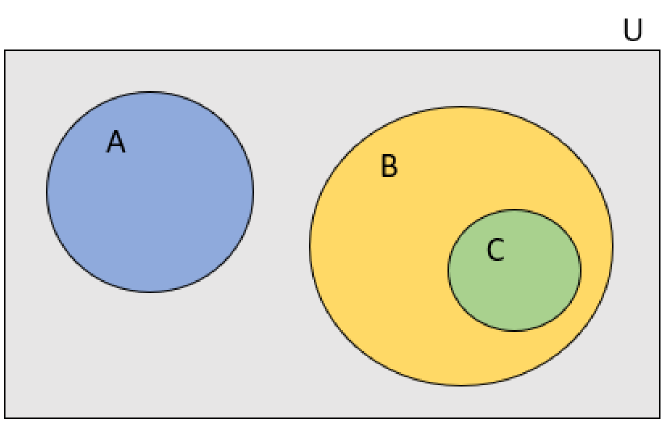


Figura 8. Ejemplo de un diagrama de Venn. Elaboración propia.

Ejemplo: Representar los conjuntos $A = \{1, 2, 3, 4, 5\}$ y $B = \{3, 7, 9\}$ a través de un diagrama de Venn.

- **Solución:** Observe la Figura 9:

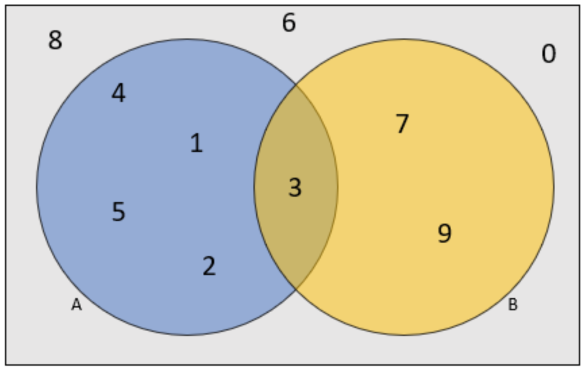


Figura 9. Conjuntos A y B, utilizando diagrama de Venn. Elaboración propia.

Sobre este diagrama de Venn se pueden hacer las siguientes afirmaciones:

- ✓ 3 está en ambos conjuntos.
- ✓ 7 y 9 están únicamente en el conjunto B.
- ✓ 1, 2, 4 y 5 están únicamente en el conjunto A.
- ✓ 0, 6 y 8 no están en los conjuntos A o B.

Los diagramas de Venn también permiten representar fácilmente las operaciones de conjuntos. A continuación se ilustra cada una de ellas utilizando diagramas de Venn:

a. **Unión:** Si se tienen dos conjuntos A y B, la unión (Figura 10) es el conjunto de todos los elementos de A o de B. Se denota por $A \cup B$.

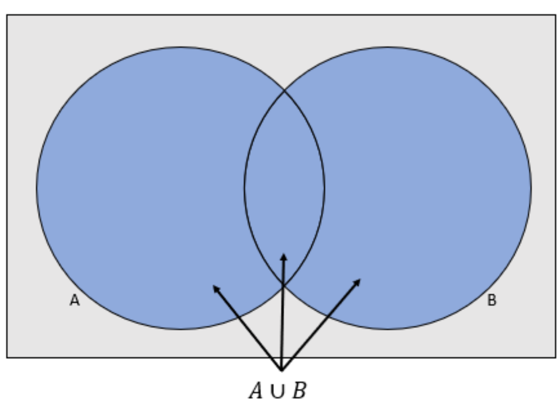


Figura 10. Unión de conjuntos, utilizando diagrama de Venn. Elaboración propia.

b. Intersección: Si se tienen dos conjuntos A y B, la intersección (Figura 11) es el conjunto de los elementos que pertenecen tanto a A como a B. Se denota por $A \cap B$.

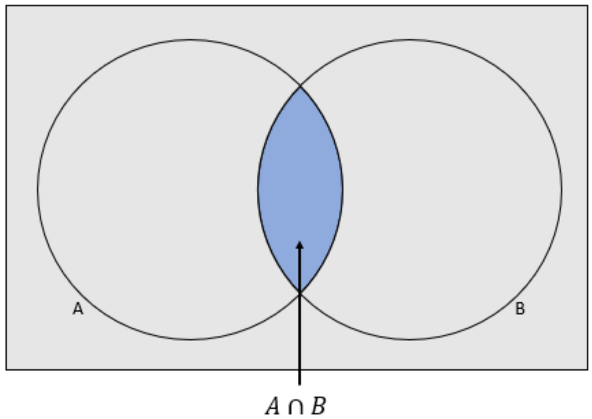


Figura 11. Intersección de conjuntos, utilizando diagrama de Venn. Elaboración propia.

c. Diferencia: Si se tienen dos conjuntos A y B, la diferencia de A menos B (Figura 12) es el conjunto de los elementos que pertenecen a A pero no pertenecen a B. Se denota por $A - B$.

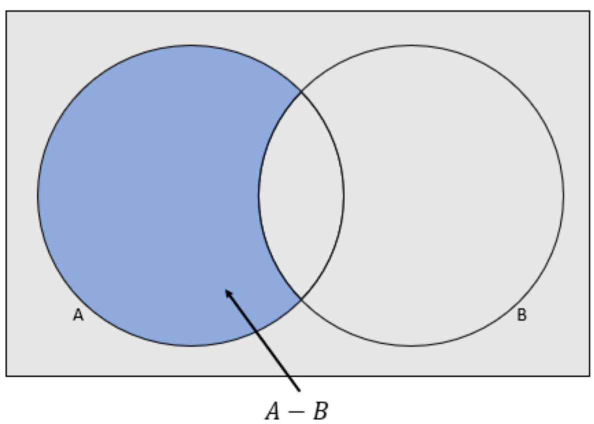


Figura 12. Diferencia de conjuntos, utilizando diagrama de Venn. Elaboración propia.

d. Complemento: Si se tiene un conjunto A, el complemento de A es el conjunto de todos los elementos que pertenecen al universo , pero que no pertenecen a A (Figura 13). Se denota por A^C, A' o $\bar{A}$.

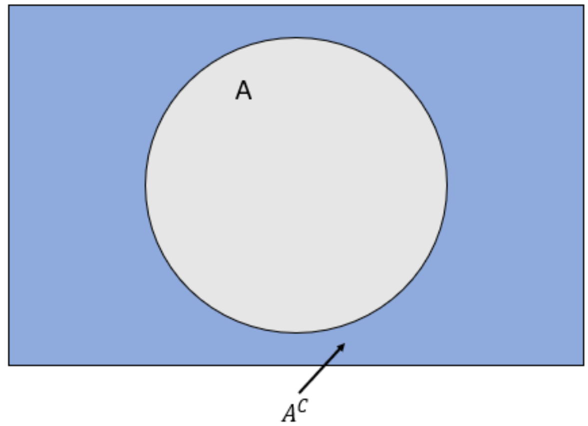


Figura 13. Complemento del conjunto A, utilizando diagrama de Venn. Elaboración propia.

Si se tiene dos conjuntos A y B, el complemento de A es el conjunto de todos los elementos que pertenecen al universo y a B, pero que no pertenecen a A (Figura 14).

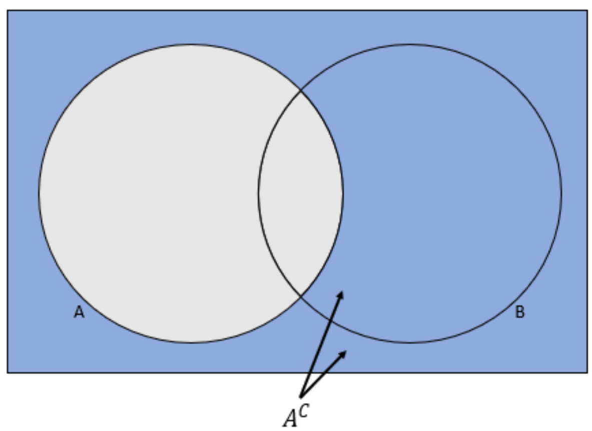


Figura 14. Complemento de dos conjuntos, utilizando diagrama de Venn. Elaboración propia.

Propiedades de los conjuntos

Los conjuntos cuentan con propiedades que permiten la simplificación de operaciones. Estas propiedades se presentan a continuación, basado en lo expuesto por (Conradie & Goranko, 2015):

a. **Doble negación:** $(A^C)^C = A$

b. **Propiedad conmutativa:**

- $A \cup B = B \cup A$

- $A \cap B = B \cap A$

c. **Propiedad asociativa:**

- $A \cup (B \cup C) = (A \cup B) \cup C$
- $A \cap (B \cap C) = (A \cap B) \cap C$

d. **Propiedad distributiva:**

- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

e. **Propiedad idempotente:**

- $A \cup A = A$
- $A \cap A = A$
- $U \cup U = U$
- $U \cap U = U$
- $\emptyset \cup \emptyset = \emptyset$
- $\emptyset \cap \emptyset = \emptyset$

f. **Propiedad de Morgan:**

- $(A \cup B \cup C)^C = A^C \cap B^C \cap C^C$
- $(A \cap B \cap C)^C = A^C \cup B^C \cup C^C$

g. **Equivalencia:** $A \cup A^C \cap B = A \cup B$

h. **Contradicción:** $A \cap A^C = \emptyset$

i. **Propiedades del complemento:**

- $A \cup A^C = U$
- $U^C = \emptyset$

- $\emptyset^C = U$

j. **Elemento universal:**

- $A \cup U = U$
- $A \cap U = A$
- $A \cup \emptyset = A$
- $A \cap \emptyset = \emptyset$
- $A \cup A \cap B = A \cap (U \cup B) = A$

Capítulo II: Sistema axiomático y su aplicación

En el Capítulo II de este documento se estudiarán el concepto de sistema axiomático y sus aplicaciones. Se definirá el concepto de sistema axiomático y se describirán sus elementos y sus propiedades. También se definirán los conceptos de equivalencia, tautología y contradicción, los cuales incluyen leyes que son muy útiles al momento de hacer demostraciones basadas en sistemas axiomáticos.

Posteriormente, se definirán dos tipos de proposiciones que se utilizan con mucha frecuencia dentro del estudio de la teoría computacional: las proposiciones condicionales y las proposiciones bicondicionales. Estas se describirán indicando su correspondiente operador y su tabla de verdad. Finalmente, se ampliarán temas estudiados en el capítulo anterior: los cuantificadores, la lógica proposicional y el sistema axiomático.

Definiciones básicas y símbolos

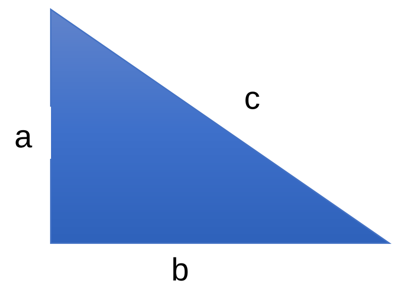
Un **sistema axiomático**, también denominado sistema formal o cálculo axiomático, es una lista de conceptos y hechos básicos a partir de los cuales se derivan otros conceptos y teoremas, a través de la definición y la deducción (Contreras Oré, 2017). Todo sistema axiomático se compone de los siguientes elementos (Garrido, 2005):

- **a.** Un **lenguaje** que se compone de una tabla de símbolos (alfabeto) y de reglas para formar fórmulas (gramática).
- **b.** Una lista de **axiomas** que son las fórmulas del sistema.
- **c.** Un conjunto de **reglas de inferencia** que permite generar nuevas fórmulas a partir de una o más fórmulas dadas. A una nueva fórmula generada se le denomina conclusión.

Un sistema axiomático no es más que un mecanismo capaz de generar una teoría, es decir, afirmaciones para un determinado dominio de problemas. Al **demostrar** una teoría, lo que se hace es validar sus proposiciones, a partir de los axiomas o de las fórmulas obtenidas como consecuencia de las reglas de inferencia. Al final de este proceso se generará un **teorema**, que es una proposición que puede ser demostrada a través de los axiomas.

Ejemplo: El teorema de Pitágoras establece que, para todo triángulo rectángulo, el cuadrado de la longitud de la hipotenusa es igual a la suma del cuadrado de los catetos. Esto se expresa con la siguiente relación:

$$a^2 + b^2 = c^2$$



Las teorías matemáticas, incluyendo la lógica, se pueden axiomatizar porque sus teoremas, por ser universalmente válidos, están incluidos en cualquier otra teoría. Además, su lenguaje y cálculo son un instrumento con el cual se pueden axiomatizar otras teorías científicas.

Los axiomas de un sistema axiomático pueden tener las siguientes propiedades, basado en lo descrito por (Contreras Oré, 2017):

- Los axiomas son **consistentes** si no hay contradicciones entre ellos; es decir, si se deduce una proposición, a la vez no se puede deducir su negación.
- Un axioma es **independiente** si no se puede deducir a partir de los otros axiomas.
- Los axiomas son **completos** si cada afirmación puede ser probada por axiomas.

Los axiomas de un sistema axiomático pueden estar expresados de manera formal o informal:

- **Axiomatización formal:** Usa un lenguaje formal y cada axioma es una cadena finita de signos en el alfabeto del lenguaje formal, siguientes reglas combinatorias que hacen de la secuencia una fórmula bien formada.

- **Axiomatización informal:** Usa una lengua natural formalizada y definiciones no ambiguas.

Los axiomas se construyen utilizando operadores lógicos. De esta manera, las proposiciones tienen valores de verdad bien definidos, el cual puede ser “verdadero” o “falso”. Por lo general, los axiomas son proposiciones compuestas, es decir, que tienen proposiciones simples unidas por operadores lógicos. Las proposiciones compuestas se representan con letras mayúsculas, mientras que las proposiciones más simples se representan con letras minúsculas.

Los valores de verdad de una proposición se pueden representar a través de una **tabla de verdad**. A continuación, de acuerdo con (Epp, 2011), se describen los valores de verdad de los operadores lógicos más comunes:

- **Negación:** Si p es una proposición, su negación es “no p”, lo cual se representa con los símbolos $\neg$ o $\sim$. El valor de verdad de $\neg p$ es el opuesto al de p, es decir, si p es verdadero, $\neg p$ es falso y si p es falso, $\neg p$ es verdadero. A continuación, en la Tabla 2, se muestra la tabla de verdad para la negación:

$\boldsymbol{p}$	$\boldsymbol{\neg p}$
V	F
F	V

Tabla 2. Tabla de verdad del operador lógico Negación.

- **Conjunción:** Si p y q son proposiciones, la conjunción de ambas es “p y q”, lo cual se representa con el símbolo $\wedge$. El valor de verdad será verdadero si y sólo si tanto p y q son verdaderas. De otro modo, si p o q son falsas o ambas son falsas, entonces $p \wedge q$ es falso. A continuación, en la Tabla 3, se muestra la tabla de verdad para la conjunción:

$\boldsymbol{p}$	$\boldsymbol{q}$	$\boldsymbol{p \wedge q}$
V	V	V
V	F	F

F	V	F
F	F	F

Tabla 3. Tabla de verdad del operador lógico Conjunción.

- **Disyunción:** Si p y q son proposiciones, la disyunción de ambas es "p o q", lo cual se representa con el símbolo ∨. El valor de verdad de $p \vee q$ será verdadero si p es verdadero, si q es verdadero o si tanto p como q son verdaderas y será falso si tanto p como q son falsos. A continuación, en la Tabla 4, se muestra la tabla de verdad para la disyunción:

$\boldsymbol{p}$	$\boldsymbol{q}$	$\boldsymbol{p \vee q}$
V	V	V
V	F	V
F	V	V
F	F	F

Tabla 4. Tabla de verdad del operador lógico Disyunción.

Equivalencia, Tautología y Contradicción

Dos proposiciones p y q son **lógicamente equivalentes** si tienen los mismos valores en su tabla de verdad (Jiménez Murillo, 2009). Esto es, si p es verdadero cuando q también es verdadero. Esto se representa como $p \equiv q$ o $p \Leftrightarrow q$. El uso de una tabla de verdad es muy útil cuando se quiere verificar si dos proposiciones son lógicamente equivalentes.

De igual manera, también se puede utilizar una serie de proposiciones de equivalencias lógicas que son útiles al momento de demostrar teoremas. A continuación, basado en (Lipschutz & Lipson, 2009), se listan algunas de estas proposiciones:

a. **Leyes de idempotencia:**
    - $p \vee p \equiv p$
    - $p \wedge p \equiv p$

b. **Leyes asociativas:**

- $(p \vee q) \vee r \equiv p \vee (q \vee r)$
- $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

c. **Leyes conmutativas:**

- $p \vee q \equiv q \vee p$
- $p \wedge q \equiv q \wedge p$

d. **Leyes distributivas:**

- $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
- $p \wedge (q \vee r) \equiv (p \wedge q) \wedge (p \wedge r)$

e. **Doble negación:** $\neg\neg p \equiv p$

f. **Leyes de De Morgan**

- $\neg(p \vee q) \equiv \neg p \wedge \neg q$
- $\neg(p \wedge q) \equiv \neg p \vee \neg q$

Una **tautología** es una proposición compuesta, representada con una letra mayúscula, que es verdadera para todos los valores de verdad de sus variables (Jiménez Murillo, 2009). Una tautología muy común es la de $p \vee \neg p$, cuya tabla de verdad se muestra a continuación en la Tabla 5:

$\boldsymbol{p}$	$\boldsymbol{\neg p}$	$\boldsymbol{p \vee \neg p}$
V	F	F
F	V	F

Tabla 5. Tabla de verdad para la tautología $p \vee \neg p$.

Las tautologías se consideran leyes y pueden utilizarse en la demostración de teoremas. A continuación, se listan algunas de las tautologías más comunes, basado en (Jiménez Murillo, 2009). Como explica (Jiménez Murillo, 2009), todas las tautologías listadas a continuación tienen la forma $P \Rightarrow Q$, lo cual se lee como "si P entonces Q".

a. **Adición:** $p \Rightarrow (p \vee q)$

b. **Simplificación:** $(p \wedge q) \Rightarrow p$

c. **Absurdo:** $(p \rightarrow q) \Rightarrow \neg p$

d. **Modus ponens:** $p \wedge (p \rightarrow q) \Rightarrow q$

e. **Modus tollens:** $(p \rightarrow q) \wedge \neg q \Rightarrow \neg p$

Una **contradicción**, también denominado absurdo, es una proposición que al evaluarla su resultado es falso para todos los valores de verdad (Jiménez Murillo, 2009). La contradicción que se utiliza con más frecuencia en la demostración de teoremas es $p \wedge \neg p$, ya que si p y $\neg p$ son verdaderas, se obtiene que $p \wedge \neg p$ también es verdadera y al ser una contradicción, entonces se concluiría que el teorema es falso. La tabla de verdad de la contradicción $p \wedge \neg p$ se muestra a continuación en la Tabla 6:

$\boldsymbol{p}$	$\boldsymbol{\neg p}$	$\boldsymbol{p \wedge \neg p}$
V	F	F
F	V	F

Tabla 6. Tabla de verdad para la tautología $p \wedge \neg p$.

Como se mencionó anteriormente, la generación de nuevas fórmulas se hace a través de las reglas de inferencia que forman parte del sistema axiomático. Una **regla de inferencia** es una forma lógica que consiste en una función que toma premisas, analiza su sintaxis y devuelve una conclusión. Como explica (Jiménez Murillo, 2009), las reglas de inferencia permiten relacionar proposiciones para obtener una nueva proposición válida en una demostración.

Las reglas de inferencia más utilizadas son las siguientes:

a. **Modus ponens (PP):** Proviene del latín "modus ponendo ponens". Esta regla de inferencia establece que si una primera afirmación implica una segunda afirmación y la primera afirmación es verdadera, entonces la segunda afirmación es verdadera. Esto se resume de la siguiente manera:

$$\begin{array}{l} p \rightarrow q \\ p \\ \hline \therefore q \end{array}$$

Ejemplo: Considere las siguientes proposiciones P y Q, y obtenga una conclusión utilizando modus ponens:

$P =$ "$Si\ hoy\ es\ lunes\ tengo\ que\ ir\ al\ trabajo$"	Premisa 1: $P \rightarrow Q$
$Q =$ "$Hoy\ es\ lunes$"	Premisa 2: P
	Conclusión: Q

b. Modus tollens (TT): Proviene del latín “modus tollendo tollens”. Esta regla de inferencia establece que si una primera afirmación implica una segunda afirmación y la segunda afirmación no es verdadera, entonces la primera afirmación no es verdadera. Esto se resume de la siguiente manera:

$$\begin{array}{l} p \rightarrow q \\ \neg q \\ \hline \therefore \neg p \end{array}$$

Ejemplo: Considere las siguientes proposiciones P y Q, y obtenga una conclusión utilizando modus tollens:

$P =$ "$Si\ el\ agua\ está\ hirviendo,$	Premisa 1: $P \rightarrow Q$
$entonces\ suelta\ vapor$"	Premisa 2: $\neg Q$
$Q =$ "$El\ agua\ no\ suelta\ vapor$"	Conclusión: $\neg P$

Otras reglas de inferencia son las que se muestran en la Tabla 7, de acuerdo con (Jiménez Murillo, 2009):

Adición	Simplificación	Silogismo disyuntivo
$\dfrac{p}{\therefore p \lor q}$	$\dfrac{p \land q}{\therefore p}$	$\dfrac{\begin{array}{c} p \lor q \\ \neg q \end{array}}{\therefore q}$
Silogismo hipotético	**Conjunción**	
$\dfrac{\begin{array}{c} p \to q \\ q \to r \end{array}}{\therefore p \to r}$	$\dfrac{\begin{array}{c} p \\ q \end{array}}{\therefore p \land q}$	

Tabla 7. Otras reglas de inferencia.

Condicionales y Bicondicionales

Una **proposición condicional** es aquella en la que se conectan dos enunciados y la verdad del segundo enunciado q, está condicionada por la verdad del primer enunciado p. Como explica (Lipschutz & Lipson, 2009), las proposiciones condicionales se utilizan mucho, principalmente en las matemáticas.

Una proposición condicional se denota utilizando el operador condicional →, de la siguiente manera:

$$p \to q$$

Donde se pueden distinguir dos partes, como describe (Epp, 2011):

- La **hipótesis** o **antecedente**, que es la parte antes del operador, es decir p.
- La **conclusión** o **consecuente**, que es la parte después del operador, es decir $\boldsymbol{q}$.

La proposición condicional $p \to q$ puede leerse como "si p entonces q", "p implica q" o "p sólo si q".

Ejemplo #1: La siguiente proposición condicional: "*Si hoy no llueve, entonces caminaré durante la tarde.*", se puede representar en notación lógica como $p \rightarrow q$ donde:

$$\textit{Si } \underbrace{\textit{hoy no llueve}}_{\boldsymbol{p}}\textit{, entonces } \underbrace{\textit{caminaré durante la tarde}}_{\boldsymbol{q}}\textit{.}$$

Ejemplo #2: La siguiente proposición condicional: "*Si el polígono tiene tres lados, entonces es un triángulo.*", se puede representar en notación lógica como $p \rightarrow q$ donde:

$$\textit{Si } \underbrace{\textit{el polígono tiene tres lados}}_{\boldsymbol{p}}\textit{, entonces } \underbrace{\textit{es un triángulo}}_{\boldsymbol{q}}\textit{.}$$

El valor de verdad de $p \rightarrow q$ será falso si p es verdadero y q es falso. De lo contrario, el valor de verdad será verdadero, incluso cuando p es falso la condicional $p \rightarrow q$ será verdadera sin importar el valor de verdad de q (Lipschutz & Lipson, 2009). La tabla de verdad de una proposición condicional $p \rightarrow q$ se muestra a continuación, en la Tabla 8:

$\boldsymbol{p}$	$\boldsymbol{q}$	$\boldsymbol{p \rightarrow q}$
V	V	V
V	F	F
F	V	V
F	F	V

Tabla 8. Tabla de verdad de una proposición condicional $p \rightarrow q$.

Existen algunas variaciones de la proposición condicional, las cuales se definen a continuación basado en las descripciones de (Epp, 2011) y (Jenkyns & Stephenson, 2018):

a) Inverso: Para una proposición condicional $p \rightarrow q$, el inverso es $\neg p \rightarrow \neg q$. Es decir, es la negación tanto del antecedente como del consecuente de la proposición original.

b) **Recíproco:** Para una proposición condicional $p \rightarrow q$, el recíproco es $q \rightarrow p$. Es decir, el consecuente pasa a ser el antecedente y el antecedente pasa a ser el consecuente.

c) **Contrapositivo:** Para un proposición $p \rightarrow q$, el contrapositivo es $\neg q \rightarrow \neg p$. Es decir, es la negación del antecedente y del consecuente del inverso de la proposición original.

A continuación, en la Tabla 9, se muestra la tabla de verdad de una proposición condicional $p \rightarrow q$ y de sus distintas variaciones, descritas anteriormente:

p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$q \rightarrow p$	$\neg p \rightarrow \neg q$	$\neg q \rightarrow \neg p$
V	V	F	F	V	V	V	V
V	F	F	V	F	V	V	F
F	V	V	F	V	F	F	V
F	F	V	V	V	V	V	V

Tabla 9. Tabla de verdad para un proposición p→q y sus variaciones.

De la tabla de verdad anterior, se puede concluir lo siguiente:

1. Una proposición condicional es lógicamente equivalente a su contrapositivo, es decir $p \rightarrow q \equiv \neg q \rightarrow \neg p$.
2. El inverso de una proposición condicional es lógicamente equivalente al recíproco de la misma proposición condicional, es decir $q \rightarrow p \equiv \neg p \rightarrow \neg q$.
3. Una proposición condicional no es lógicamente equivalente ni a su recíproco ni a su inverso.

Ejemplo: Obtener el inverso, el recíproco y contrapositivo de la siguiente proposición condicional: *"Si el polígono tiene tres lados, entonces es un triángulo"*.

- **Solución:** De la proposición condicional se tiene que:

$p = el\ polígono\ tiene\ tres\ lados$

$q = es\ un\ triángulo$

$\neg p = el\ polígono\ no\ tiene\ tres\ lados$

$\neg q = no\ es\ un\ triángulo$

A partir de lo anterior, se obtiene lo siguiente para la proposición condicional dada:

Inverso ⟶ $\neg p \rightarrow \neg q$

*Si el polígono **no** tiene tres lados, entonces **no** es un triángulo.*

Recíproco ⟶ $q \rightarrow p$

Si es un triángulo, entonces el polígono tiene tres lados.

Contrapositivo ⟶ $\neg q \rightarrow \neg p$

Si **no** es un triángulo, entonces el polígono **no** tiene tres lados.

Una **proposición bicondicional** conecta dos enunciados p y q, donde p es verdadero **si y sólo si** q es verdadero. De igual manera, p es falso **si y sólo si** q es falso.

Se denota utilizando el operador bicondicional ↔, de la siguiente manera:

$$p \leftrightarrow q$$

La proposición bicondicional $p \leftrightarrow q$ puede leerse como "p si y sólo si q".

Ejemplo #1: La siguiente proposición condicional: "*El estudiante pasa la materia, si y sólo si obtiene A en los tres parciales.*", se puede representar lógicamente como $p \leftrightarrow q$ donde:

$\underbrace{\textit{El estudiante pasa la materia}}_{\boldsymbol{p}}$*, si y sólo si* $\underbrace{\textit{obtiene A en los tres parciales}}_{\boldsymbol{q}}$.

Ejemplo #2: La siguiente proposición condicional: "*Debo ir al trabajo, si y sólo si mañana es lunes.*", se puede representar lógicamente como $p \leftrightarrow q$ donde:

$$\underbrace{\textit{Debo ir al trabajo}}_{p}\textit{, si y sólo si }\underbrace{\textit{mañana es lunes}}_{q}.$$

El valor de verdad de $p \leftrightarrow q$ será verdadero si tanto p como q tienen los mismos valores de verdad, sino entonces $p \leftrightarrow q$ será falso. La tabla de verdad de una proposición bicondicional $p \leftrightarrow q$ se muestra a continuación, en la Tabla 10:

p	q	$p \leftrightarrow q$
V	V	V
V	F	F
F	V	F
F	F	V

Tabla 10. Tabla de verdad de una proposición bicondicional $p \leftrightarrow q$.

Como explican (Epp, 2011) y (Jenkyns & Stephenson, 2018), la proposición bicondicional $p \leftrightarrow q$ es lógicamente equivalente a la conjunción de dos expresiones condicionales, de modo que $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$. Esto se puede comprobar con la tabla de verdad que se presenta a continuación, en la Tabla 11:

p	q	$p \rightarrow q$	$q \rightarrow p$	$p \leftrightarrow q$	$(p \rightarrow q) \wedge (q \rightarrow p)$
V	V	V	V	V	V
V	F	F	V	F	F
F	V	V	F	F	F
F	F	V	V	V	V

Tabla 11. Demostración de $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$.

Cuantificadores

Un **cuantificador** convierte una declaración simbólica acerca de cualquier elemento del universo en una declaración acerca del universo (Stein, Drysdale, & Bogart, 2011). Es decir, especifican *cuántos* elementos del universo tienen una determinada propiedad, en lugar de indicar cuáles objetos tienen dicha propiedad. En este sentido, como explica (Jiménez Murillo, 2009), los cuantificadores se refieren a la cantidad de elementos para los cuales un predicado dado es verdadero.

Se pueden distinguir dos tipos de cuantificadores, los cuales indican que el predicado es verdadero o falso, ya sea para *todos los posibles objetos* o para *algunos objetos* del universo. Estos dos cuantificadores se describen a continuación, basado en lo explicado por (Stein et al., 2011) y (Jiménez Murillo, 2009):

a) Cuantificador universal: Afirma que una declaración acerca de una variable es verdadera para todos los valores de la variable en el universo. El símbolo utilizado para denotar este cuantificador es ∀ y se lee "para cada" o "para todo". Una declaración universal es de la forma:

$\forall x \in D, P(x)$ → Donde:

- $P(x)$ es un predicado
- D es el dominio de x

Esta declaración es verdadera si y sólo si $P(x)$ es verdadera para cada x en D. Por otro lado, esta declaración es falsa si y sólo si $P(x)$ es falsa para al menos un x en D.

Ejemplo #1: Considere la siguiente declaración $\forall x \in D, x^2 \geq x$, donde $D = \{2, 4, 6, 8\}$. Indique si la declaración es verdadera o falsa.

- **Solución:** Reemplazando la variable del predicado de la declaración con los valores de D en la variable, se obtiene lo siguiente:

$$2^2 \geq 2 \rightarrow 4 \geq 2,$$

$$4^2 \geq 4 \rightarrow 16 \geq 4,$$

$$6^2 \geq 6 \rightarrow 36 \geq 6,$$

$$8^2 \geq 8 \rightarrow 64 \geq 8$$

$x^2 \geq x$ es verdadero para cada x en D. Por lo tanto, la declaración $\forall \boldsymbol{x} \in \boldsymbol{D}, \boldsymbol{x}^2 \geq \boldsymbol{x}$ **es verdadera**.

Ejemplo #2: Considere la siguiente declaración $\forall n \in N, (n+4) > 5$, donde N corresponde al conjunto de los números naturales. Indique si la declaración es verdadera o falsa.

- **Solución:** Reemplazando la variable del predicado de la declaración con algunos de los valores de N en la variable, se obtiene lo siguiente:

$$(1+4) > 5 \rightarrow 5 = 5,$$

$$(2+4) > 5 \rightarrow 6 > 5,$$

$$(3+4) > 5 \rightarrow 7 > 5,$$

$$(4+4) > 5 \rightarrow 8 > 5$$

$(n+4) > 5$ es falso para $n = 1$, aunque no ocurre lo mismo con los demás valores. Esto quiere decir que $(n+4) > 5$ es falso para al menos un elemento de N. Por lo tanto, la declaración $\forall \boldsymbol{n} \in \boldsymbol{N}, (\boldsymbol{n}+\boldsymbol{4}) > \boldsymbol{5}$ **es falsa**.

b) Cuantificador existencial: Afirma que cierto elemento del universo existe. El símbolo utilizado para denotar este cuantificador es $\exists$ y se lee "existe" o "hay algunos". por lo que una declaración existencial es de la forma:

$\exists x \in D, tal\ que\ P(x)$ $\longrightarrow$ Donde:

- $P(x)$ es un predicado
- D es el dominio de x

Esta declaración es verdadera si y sólo si $P(x)$ es verdadera para al menos una x en D. Por otro lado, esta declaración es falsa si y sólo si $P(x)$ es falsa para toda x en D.

Ejemplo #1: Considere la siguiente declaración $\exists n \in Z^+, tal\ que\ n^2 = n$, donde Z^+ corresponde al conjunto de los números enteros positivos. Indique si la declaración es verdadera o falsa.

- **Solución:** Reemplazando la variable del predicado de la declaración con algunos de los valores de Z^+, se obtiene lo siguiente:

$$1^2 = 1 \rightarrow 1 = 1,$$

$$2^2 = 4 \rightarrow 4 \neq 2,$$

$$3^2 = 9 \rightarrow 9 \neq 3,$$

$$4^2 = 16 \rightarrow 16 \neq 4$$

$n^2 \geq n$ es verdadero para $n = 1$, pero no para los demás valores de Z^+. Esto quiere decir que $n^2 \geq n$ es verdadero para al menos un elemento de Z^+. Por lo tanto, la declaración $\exists \boldsymbol{n} \in \boldsymbol{Z^+}, \boldsymbol{tal\ que\ n^2 = n}$ **es verdadera**.

Ejemplo #2: Considere la siguiente declaración $\exists n \in I, tal\ que\ n^2 = n$, donde $I = \{2,3,4\}$. Indique si la declaración es verdadera o falsa.

- **Solución:** Reemplazando el predicado de la declaración con los valores de I en la variable, se obtiene lo siguiente:

$$2^2 = 2 \rightarrow 4 \neq 2,$$

$$3^2 = 3 \rightarrow 9 \neq 3,$$

$$4^2 = 4 \rightarrow 16 \neq 4$$

$n^2 = n$ es falso para toda n en N. Por lo tanto, la declaración $\exists \boldsymbol{n} \in \boldsymbol{I}, \boldsymbol{tal\ que\ n^2 = n}$ **es falsa**.

Cálculo proposicional

La lógica proposicional se encarga de estudiar cómo las proposiciones se relacionan unas con otras y si cada proposición es una declaración verdadera o falsa (O'Regan, 2013) (Levin, 2018). En este sentido, el **cálculo proposicional**, también denominado *lógica de orden 0*, estudia el álgebra de las proposiciones, por lo que no admite el uso de predicados y por lo tanto, no se usan variables ni cuantificadores.

Como explica (Del Callejo-Canal, Del Callejo-Canal, & Canal-Martínez, 2016), el objetivo del cálculo proposicional es aplicar las reglas de inferencia y de equivalencia para realizar una *deducción* y finalmente obtener una *conclusión* o *consecuencia lógica*, a partir de un conjunto de premisas (conjunto de proposiciones iniciales). Las tablas de verdad también pueden utilizarse dentro del cálculo proposicional para demostrar la validez de proposiciones; sin embargo, este método no se recomienda para casos en los que hay un gran número de variables proposicionales, ya que hay 2^n entradas en la tabla de verdad para n variables proposicionales (O'Regan, 2013).

Ejemplo #1: Considere las siguientes premisas y demuestre la proposición R:

(1) $P \rightarrow Q$	P
(2) $Q \rightarrow R$	P
(3) P	P

- **Solución:** Aplicando la regla de inferencia modus ponens (PP), se puede llegar a R en dos pasos:

(1) $P \rightarrow Q$	P
(2) $Q \rightarrow R$	P
(3) P	P
(4) Q	PP 1,3
(5) R	PP 2,4

Indica la regla de inferencia utilizada y la línea de donde proviene

Ejemplo #2: Considere las siguientes premisas y demuestre la proposición U:

(1) $\neg P \to Q$	P
(2) $Q \to R$	P
(3) $\neg P$	P
(4) $R \to \neg S$	P
(5) $\neg S \to T$	P
(6) $T \to U$	P

- **Solución:** La demostración se presenta a continuación:

(1) $\neg P \to Q$	P
(2) $Q \to R$	P
(3) $\neg P$	P
(4) $R \to \neg S$	P
(5) $\neg S \to T$	P
(6) $T \to U$	P
(7) Q	PP 1,3
(8) R	PP 2,7
(9) $\neg S$	PP 4,8
(10) T	PP 5,9
(11) U	PP 6,10

Ejemplo #3: Considere las siguientes premisas y demuestre $\neg\neg R$:

(1) $P \to Q$	P
(2) $\neg Q$	P
(3) $\neg P \to Q$	P

- **Solución:** La demostración se presenta a continuación:

(1) $P \rightarrow Q$	P
(2) $\neg Q$	P
(3) $\neg P \rightarrow Q$	P
(4) $\neg P$	TT 1,2
(5) R	PP 3,4
(6) $\boxed{\neg\neg R}$	DN 5

Las reglas de inferencia utilizadas en este ejemplo son modus ponens (PP), modus tollens (TT) y la doble negación (DN).

Ejemplo #4: Demostrar a través de una tabla de verdad que:

$$(P \wedge Q) \vee (Q \wedge \neg R) \vdash (Q \vee R)$$

- **Solución:** A continuación se presenta la tabla de verdad:

P	Q	R	$\neg R$	$P \wedge Q$	$Q \wedge \neg R$	$(P \wedge Q) \vee (Q \wedge \neg R)$	$(Q \vee R)$
V	V	V	F	V	F	**V**	**V**
V	V	F	V	V	V	**V**	**V**
V	F	V	F	F	F	F	V
V	F	F	V	F	F	F	F
F	V	V	F	F	F	F	V
F	V	F	V	F	V	**V**	**V**
F	F	V	F	F	F	F	V
F	F	F	V	F	F	F	F

Las líneas señaladas de color azul indican las entradas en las que tanto $(P \wedge Q) \vee (Q \wedge \neg R$ como $(Q \vee R)$ son verdaderas. Por lo tanto, se demuestra que $(P \wedge Q) \vee (Q \wedge \neg R) \vdash (Q \vee R)$.

Nota: La notación $A \vdash B$ se puede leer como “B se deduce a partir de A”, “B es demostrable a partir de A” o “B es consecuencia lógica de A”.

Sistema axiomático

Recordemos que un sistema axiomático se compone de un lenguaje, de axiomas y de reglas de inferencia, y que a partir de los axiomas se derivan otros razonamientos y proposiciones. Utilizando los axiomas a través de una demostración se pueden obtener teoremas.

Ejemplo: Demostrar que $(P \to P)$ es un teorema del sistema axiomático A, en el cual la única regla de inferencia es el modus ponens (PP) y tiene los siguientes axiomas:

$$\boldsymbol{Ax1.}\big(P \to (Q \to P)\big)$$

$$\boldsymbol{Ax2.}\big((P \to (Q \to R)\big) \to ((P \to Q) \to (P \to R)))$$

$$\boldsymbol{Ax3.}((\neg Q \to \neg P) \to ((\neg Q \to P) \to Q))$$

- **Solución:** A continuación está la demostración:

(1) $\Big((P \to \big((P \to P) \to P\big)\Big) \to \Big(\big(P \to (P \to P)\big) \to (P \to P)\big)\Big)$	$Ax2$
(2) $\big(P \to \big((P \to P) \to P\big)\big)$	$Ax1$
(3) $\big((P \to (P \to P)\big) \to (P \to P))$	PP 2,1
(4) $(P \to (P \to P))$	$Ax1$
(5) $(P \to P)$	PP 4,3

Por lo tanto, se concluye que $\vdash_A (P \to P)$, es decir, que $(P \to P)$ es un teorema de A.

Capítulo III: Elementos fundamentales de la teoría de conmutación

En el Capítulo III de este documento se estudiarán conceptos y elementos fundamentales de la teoría de conmutación. Esto incluye el estudio del álgebra booleana, sus elementos, propiedades y teoremas los cuales serán muy útiles para la resolución de problemas. El álgebra booleana tiene numerosas aplicaciones en las ciencias computacionales y por lo tanto, es de gran importancia comprender este tema. Adicionalmente, se describirá más detalladamente el concepto de función lógica y se estudiarán las diversas formas en las que se puede representar una función lógica, mostrando.

Por último, se describirán las funciones básicas que permiten expresar funciones más complejas, dentro de la lógica booleana. Para cada una de ellas, se presentará tanto la función que la representa como su correspondiente tabla de verdad. Posteriormente, se describirá en qué consisten los conjuntos funcionalmente completos y cómo estos pueden representar otras funciones.

Álgebra booleana

El **álgebra booleana**, también llamada álgebra de Boole, en informática y matemática, es una estructura algebraica que esquematiza las operaciones lógicas.

El álgebra booleana fue desarrollada por George Boole y en su libro *“An Investigation of the Laws of Thought”,* publicado en 1854, describió las herramientas para lograr que proposiciones lógicas puedan manipularse de manera algebraica (Jiménez Murillo, 2009). Sin embargo, no fue hasta 1938 cuando esto empezó a tener una aplicación directa, gracias a que la compañía estadounidense de teléfonos *“Bell”*, utilizó estas herramientas para analizar los circuitos de su red telefónica.

Ese mismo año, Claude E. Shannon (Figura 15), desarrolló la denominada **álgebra de conmutación** a partir del álgebra booleana y demostró las importantes aplicaciones de esta última en el diseño y representación de circuitos lógicos de control basados en interruptores.

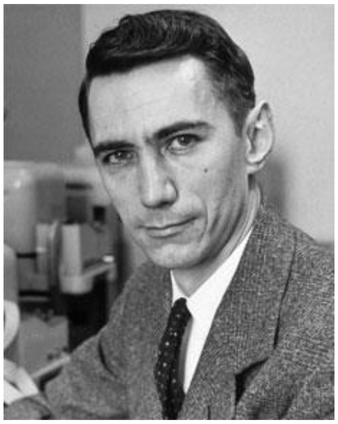


Figura 15. Claude E. Shannon. Fuente: History Computer.

El álgebra booleana y los circuitos lógicos de control tienen numerosas aplicaciones, por lo que son de gran importancia. Entre estas aplicaciones se pueden mencionar las computadoras, los robots y los sistemas telefónicos (Jiménez Murillo, 2009).

El estudio de los circuitos lógicos abarca el análisis de las señales, las cuales representan información y pueden interpretarse como un valor o una cadena de valores. En este sentido, las señales pueden ser de dos tipos: analógicas y digitales, como se muestra en la Figura 16.

Figura 16. Tipos de señales. Elaboración propia.

Una señal binaria, que es el tipo de señal que se analiza dentro del estudio del álgebra booleana, es un tipo de señal digital que sólo puede tomar dos posibles valores: 1 o 0, verdadero o falso, conectado o desconectado (Jiménez Murillo, 2009).

Como explica (Katz & Borriello, 2005), el álgebra booleana consiste en un conjunto B, que contiene:

- Dos elementos → 0 (falso) y 1 (verdadero)
- Dos operaciones binarias → AND o producto lógico $(\cdot)$ y OR o suma lógica $(+)$.
- Una operación unaria → NOT o negación, representado como $(-)$ o $(')$. También se denomina complemento.

Dentro de la lógica booleana existen una serie de definiciones básicas:

a) **Variable:** Símbolo que puede ser sustituido por un elemento del conjunto $B = \{0, 1\}$.

b) **Constante:** Es un valor que pertenece al conjunto $\{0, 1\}$.

c) **Expresión:** Se compone de variables, constantes y operadores.

d) **Función:** Una función booleana de n variables $f(x_1, x_2, \dots, x_n)$, es una expresión que mapea f a un valor del conjunto booleano B.

e) **Literal:** Es una variable o su complemento.

Estas definiciones básicas se exponen con el ejemplo mostrado en la Figura 17:

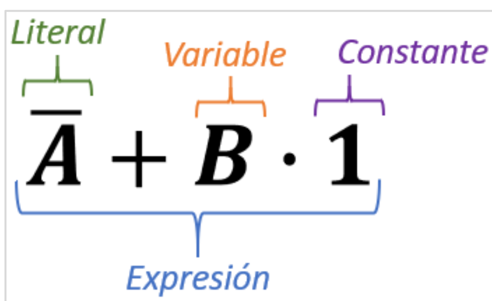


Figura 17. Definiciones básicas del álgebra booleana. Elaboración propia.

Teoremas del álgebra de Boole

El álgebra de Boole se rige por propiedades y reglas denominadas leyes o postulados. Estas permiten simplificar expresiones lógicas o transformarlas en expresiones equivalentes (Jiménez Murillo, 2009).

Todas las expresiones booleanas se caracterizan por su dualidad. El *dual* de una expresión booleana se basa en la expresión original reemplazando las operaciones AND

por operaciones OR y viceversa, y reemplazando las constantes lógicas 0 por 1 y viceversa (Katz & Borriello, 2005).

Ejemplo: El dual de la expresión booleana $\boldsymbol{A+B\cdot 1}$ es $\boldsymbol{A\cdot B+0}$, como se muestra en la Figura 18:

$$\overbrace{\boldsymbol{A+B\cdot 1}}^{\text{Expresión original}} \longrightarrow \overbrace{\boldsymbol{A\cdot B+0}}^{\text{Dual}}$$

Figura 18. Ejemplo de la dualidad de una expresión booleana. Elaboración propia.

La dualidad es un teorema fundamental del álgebra booleana, ya que cualquier declaración que sea cierta para una expresión booleana, también será cierta para su dual.

A continuación se describen las propiedades de las operaciones del álgebra booleana, basado en (Jiménez Murillo, 2009):

1. **Existencia de neutros**
   - $A+0=A$
   - $A\cdot 1=A$
2. **Conmutativa**
   - $A+B=B+A$
   - $A\cdot B=B\cdot A$
3. **Asociativa**
   - $A+(B+C)=(A+B)+C$
   - $A\cdot(B\cdot C)=(A\cdot B)\cdot C$
4. **Distributiva**
   - $A+(B\cdot C)=(A+B)\cdot(A+C)$
   - $A\cdot(B+C)=(A\cdot B)+(A\cdot C)$
5. **Existencia de complementos**
   - $A+\bar{A}=1$
   - $A\cdot\bar{A}=0$

A partir de las propiedades se pueden demostrar los siguientes teoremas, basado en (Jiménez Murillo, 2009) y (Katz & Borriello, 2005):

a) **T1. Idempotencia**
   - $A + A = A$
   - $A \cdot A = A$

b) **T2. Identidad de los elementos 0 y 1**
   - $A + 1 = 1$
   - $A + 0 = A$
   - $A \cdot 0 = 0$
   - $A \cdot 1 = A$

c) **T3. Absorción**
   - $A + (A \cdot B) = A$
   - $A \cdot (A + B) = A$

d) **T4. Complemento de 0 y 1**
   - $0' = 1$
   - $1' = 0$

e) **T5. Involución**
   - $(A')' = A$

f) **T6. Leyes de DeMorgan**
   - $(A + B)' = A' \cdot B'$
   - $(A \cdot B)' = A' + B'$

Representación de funciones lógicas

Una **función lógica**, también denominada **función booleana**, describe las operaciones necesarias para que determinadas entradas de un circuito produzcan una salida que sólo puede tomar los valores de 0 y 1 (LaMeres, 2017).

El valor de una función lógica es equivalente al de una expresión de álgebra booleana y dependerá de los valores asignados a las variables, así como de las operaciones que componen a la función. Como explica (LaMeres, 2017), cada función lógica describe una sola salida de un circuito; sin embargo, si el circuito tiene múltiples salidas se necesitará una función lógica para cada salida.

Una función lógica puede denotarse con una letra mayúscula de la forma $\boldsymbol{F(A,B)}$, $\boldsymbol{F_{A,B}}$ o simplemente $\boldsymbol{F}$. En los dos primeros casos se indican las variables o entradas de la función. Las operaciones que relacionan a las variables lógicas (entradas) en una función son las tres operaciones que forman parte del álgebra booleana, las cuales se describen a continuación:

a) **Producto lógico (·):** Un producto lógico tendría la forma $\boldsymbol{F = A \cdot B}$, donde el valor de F será 0 o 1 según los valores de las variables A y B. Como expone (Jiménez Murillo, 2009), de acuerdo con el álgebra booleana se tiene que:

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

b) **Suma lógica (+):** Una suma lógica tendría la forma $\boldsymbol{F = A + B}$, donde el valor de F será 0 o 1 según los valores de las variables A y B. Como presenta (Jiménez Murillo, 2009), de acuerdo con el álgebra booleana se tiene que:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

c) **Negación (–):** La negación invierte el valor de una variable, por lo que si una variable tiene el valor de 0 su negación tomará el valor de 1 y viceversa. Una función con una variable negada tendría la forma $\boldsymbol{F = \overline{A}}$. Como expone (Jiménez Murillo, 2009), de acuerdo con el álgebra booleana se tiene que:

$$0 = 1$$

$$1 = 0$$

Las funciones lógicas pueden representarse de diversas formas, entre las que se pueden mencionar: tablas de verdad, formas canónicas, diagramas de compuertas y mapas de Karnaugh.

Tabla de verdad

Una **tabla de verdad** es una especificación formal que describe el comportamiento exacto de un circuito para todas las posibles entradas, especificando las salidas para cada combinación de entradas (Kumar Sarkar, Kumar De, & Sarkar, 2014). En una tabla de verdad se despliegan los valores de todas las combinaciones posibles de las variables y el valor que se le asocia a la función lógica.

Una tabla de verdad tendrá $\mathbf{2^n}$ filas, donde n es la cantidad de variables; mientras que la cantidad columnas será una más que la cantidad de variables formando así una función lógica (Kumar Sarkar et al., 2014). Por otro lado, como explica (LaMeres, 2017), los códigos binarios dentro de la tabla de verdad se colocan en orden ascendente simulando un conteo binario empezando desde 0.

A continuación, en la Tabla 12, se muestran ejemplos de cómo sería la tabla de verdad para casos donde hay una, dos y tres variables:

<table>
<tr><th>1 variable
$2^1 = 2\ filas$</th><th>2 variables
$2^2 = 4\ filas$</th><th>3 variables
$2^3 = 8\ filas$</th></tr>
<tr>
<td><table><tr><th>A</th></tr><tr><td>0</td></tr><tr><td>1</td></tr></table></td>
<td><table><tr><th>A</th><th>B</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table></td>
<td><table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td>
</tr>
</table>

Tabla 12. Tabla de verdad para una, dos y tres variables.

Formas canónicas

Una **forma canónica** es un término estándar con el cual se representa una función y permite comparar funciones booleanas que están expresadas en términos algebraicos (Katz & Borriello, 2005).

Se distinguen dos formas canónicas, las cuales se definen a continuación basado en las descripciones de (Katz & Borriello, 2005) y (Marcovitz, 2010):

a) Suma de productos (SOP): También denominada **forma canónica disyuntiva**, es aquella expresión en la que una o más productos lógicos de uno o más literales, están conectados por operadores OR.

Ejemplo: $F = \bar{A}\bar{B} + A\bar{B} + AB$

Una expresión SOP se obtiene tomando aquellas filas de una tabla de verdad en las que la función es igual a 1. Las variables cuyo valor es 0 en estas filas se colocan como complemento en la función.

b) Producto de sumas (POS): También denominada **forma canónica conjuntiva**, es aquella expresión en la que una o más sumas lógicas de uno o más literales, están conectados por operadores AND.

Ejemplo: $F = (\bar{A} + \bar{B}) \cdot (A + \bar{B}) \cdot (A + B)$

Una expresión POS se obtiene tomando aquellas filas de una tabla de verdad en las que la función es igual a 0. Las variables cuyo valor es 1 en estas filas se colocan como complemento en la función.

Conversión de una forma a otras

Las funciones lógicas pueden representarse de diversas maneras y debido a esto, es posible pasar de una representación a otra. A continuación, se presentan diferentes ejemplos en donde se describirá cómo convertir de:

- Función a tabla de verdad
- Función en forma canónica a tabla de verdad
- Tabla de verdad a función en forma canónica

Ejemplo #1: Obtener la tabla de verdad de la siguiente función:

$$F_{A,B,C} = A\bar{B} + BC + \bar{A}\bar{B}\bar{C}$$

- **Solución:** Se observa que la función tiene tres variables: A, B y C. Tomando esto en cuenta, se sabe que la tabla de verdad tendrá 8 filas porque $2^3 = 8$. Las tres primeras columnas de la tabla corresponden a cada una de las variables de la función y las columnas restantes corresponden a los términos de la función y al resultado de la función:

Variables o entradas			Términos de la función			Función
A	B	C	$A\bar{B}$	BC	$\bar{A}\bar{B}\bar{C}$	F

$2^3 = 8\ filas$

Observe que en este caso:

a) Todos los términos son productos lógicos.

b) Las variables que estén negadas tomarán el valor contrario al que tienen en las columnas de variables. Por ejemplo, A en la primera fila tiene un valor de 0, su valor en el tercer término será 1 porque está negada o complementada en ese término.

c) El valor de la función es la suma lógica de cada uno de los términos de la función dada.

Considerando esto, es importante recordar las reglas del álgebra booleana para poder realizar las operaciones para obtener el valor de la función:

Producto lógico

$$\boxed{\begin{array}{l} 0 \cdot 0 = 0 \\ 0 \cdot 1 = 0 \\ 1 \cdot 0 = 0 \\ 1 \cdot 1 = 1 \end{array}}$$

Suma lógica

$$\boxed{\begin{array}{l} 0 + 0 = 0 \\ 0 + 1 = 1 \\ 1 + 0 = 1 \\ 1 + 1 = 1 \end{array}}$$

Negación

$$\boxed{\begin{array}{l} 0 = 1 \\ 1 = 0 \end{array}}$$

La tabla de verdad para $F_{A,B,C} = A\bar{B} + BC + \bar{A}\bar{B}\bar{C}$ sería la siguiente:

A	B	C	$A\bar{B}$	BC	$\bar{A}\bar{B}\bar{C}$	F
0	0	0	0	0	1	1
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	0	1
1	0	1	1	0	0	1
1	1	0	0	0	0	0
1	1	1	0	1	0	1

Ejemplo #2: Obtener la tabla de verdad de la siguiente función:

$$F_{A,B,C} = \bar{A}B + (A + \bar{C})(B + C)$$

- **Solución:** Se observa que la función tiene tres variables: A, B y C. Similar al ejemplo anterior, la tabla de verdad tendrá 8 filas porque $2^3 = 8$. Las tres primeras columnas de la tabla corresponden a cada una de las variables de la función y las columnas restantes corresponden a los términos de la función y al resultado de la función.

  Observe que en este caso:

  a) El primer término es un producto lógico y el tercer y cuarto término son sumas lógicas dentro de paréntesis. Estos paréntesis indican que ambos términos se multiplican y por lo tanto, se tiene un producto lógico, como se muestra a continuación:

Sumas lógicas

$$F_{A,B,C} = \bar{A}B + \underbrace{(A+\bar{C})(B+C)}_{\text{Producto lógico}}$$

b) El valor de la función resulta de la suma lógica entre el primer término y el producto lógico formado entre el segundo y tercer término.

Al igual que en el ejemplo anterior, los resultados de cada uno de estos término se obtiene siguiendo las reglas del álgebra booleana. Considerando esto, la tabla de verdad para $F_{A,B,C} = \bar{A}B + (A+\bar{C})(B+C)$ sería la siguiente:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{C}$	$\boldsymbol{\bar{A}B}$	$\boldsymbol{A+\bar{C}}$	$\boldsymbol{B+C}$	$\boldsymbol{(A+\bar{C})(B+C)}$	$\boldsymbol{F}$
0	0	0	0	1	0	0	0
0	0	1	0	0	1	0	0
0	1	0	1	1	1	1	1
0	1	1	1	0	1	0	1
1	0	0	0	1	0	0	0
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	0	1	1	1	1

Ejemplo #3: Obtener la tabla de verdad de la siguiente función expresada como una suma de productos:

$$S = \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC$$

- **Solución:** Como en los ejemplos anteriores, se tienen tres variables y por consiguiente, la tabla de verdad tendrá 8 filas. La función dada está expresada como una suma de productos (SOP) y por lo tanto se debe recordar que:

  a) Se toman aquellas filas en las que la función tiene un valor de 1.

  b) En aquellas filas en las que la función tiene un valor de 1, las variables que tienen el valor de 0, aparecerán negadas o complementadas en la función.

Por lo tanto, la tabla de verdad de $S = \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC$ es la siguiente:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{C}$	$\boldsymbol{S}$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

A continuación, se muestran los términos de la función y sus correspondientes filas:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{C}$	$\boldsymbol{S}$	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	**1**	$\bar{A}BC$
1	0	0	**1**	$A\bar{B}\bar{C}$
1	0	1	0	
1	1	0	**1**	$AB\bar{C}$
1	1	1	**1**	ABC

Ejemplo #4: Obtener la tabla de verdad de la siguiente función expresada como un producto de sumas:

$$S = (A + B + C)(A + B + \bar{C})(A + \bar{B} + \bar{C})(\bar{A} + B + C)$$

- **Solución:** Como en los ejemplos anteriores, se tienen tres variables y por consiguiente, la tabla de verdad tendrá 8 filas. La función dada está expresada como una suma de productos (SOP) y por lo tanto se debe recordar que:

  a) Se toman aquellas filas en las que la función tiene un valor de 0.

  b) En aquellas filas en las que la función tiene un valor de 1, las variables que tienen el valor de 1, aparecerán negadas o complementadas en la función.

  Por lo tanto, la tabla de verdad de $S = (A + B + C)(A + B + \bar{C})(A + \bar{B} + \bar{C})(\bar{A} + B + C)$ es la siguiente:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{C}$	$\boldsymbol{S}$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

  A continuación, se muestran los términos de la función y sus correspondientes filas:

A	B	C	S	
0	0	0	**0**	$(A + B + C)$
0	0	1	**0**	$(A + B + \bar{C})$
0	1	0	1	
0	1	1	**0**	$(A + \bar{B} + \bar{C})$
1	0	0	**0**	$(\bar{A} + B + C)$
1	0	1	1	
1	1	0	1	
1	1	1	1	

Ejemplo #5: Obtener las formas canónicas (suma de productos y producto de sumas), a partir de la siguiente tabla de verdad:

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

- **Solución:** Recordando las reglas para obtener cada una de las formas canónicas (descritas en el **Ejemplo #3** y el **Ejemplo #4**), la suma de productos y el producto de sumas de la tabla de verdad dada serían:

  SOP: $F(X, Y, Z) = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + \bar{X}YZ + X\bar{Y}\bar{Z} + X\bar{Y}Z$

  POS: $F(X, Y, Z) = (X + Y + Z)(\bar{X} + \bar{Y} + Z)(\bar{X} + \bar{Y} + \bar{Z})$

Funciones básicas

Generalmente, las funciones lógicas son muy complejas. Sin embargo, estas siempre se componen de una combinación de las funciones lógicas básicas, que son las mismas operaciones básicas del álgebra booleana. Como describe (Garrido, 2005), todas las funciones lógicas pueden expresarse en términos de conjunción (AND), disyunción (OR) y negación (NOT).

(Vingron, 2012) explica que se pueden distinguir funciones lógicas de una entrada y de dos entradas. La siguiente función básica tienen una entrada:

- **NOT ($-$ o $'$):** En esta función representada como $\boldsymbol{F = \overline{A}}$, el valor de la salida es la negación del valor de la entrada, es decir se invierte el valor de la variable. Dicho de otro modo, si la entrada tiene un valor de 0, la salida tendrá un valor de 1; mientras que si la entrada tiene un valor de 1, la salida tendrá un valor de 0. La tabla de verdad de esta función se muestra en la Tabla 13:

$\boldsymbol{A}$	$\boldsymbol{F = \overline{A}}$
0	1
1	0

Tabla 13. Tabla de verdad de la función básica NOT.

Las siguientes funciones básicas tienen como mínimo dos entradas, sin embargo siempre tienen una sola salida:

- **AND ($\cdot$):** En esta función representada como $\boldsymbol{F = A \cdot B}$ o como $\boldsymbol{F = AB}$, el valor de la salida será igual a 0 si al menos una de las entradas tiene el valor de 0; de lo contrario, si todas las entradas tienen el valor de 1, entonces el valor de la salida será 1. Esto se demuestra en la tabla de verdad de esta función, la cual se muestra a continuación en la Tabla 14:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{F = A \cdot B}$
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 14. Tabla de verdad de la función básica AND.

- **OR (+):** En esta función representada como $\boldsymbol{F = A + B}$, el valor de la salida será igual a 1 si al menos una de las entradas tiene el valor de 1; de lo contrario, si todas las entradas tienen el valor de 0, entonces el valor de la salida será 0. Esto se demuestra en la tabla de verdad de esta función, la cual se muestra a continuación en la Tabla 15:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{F = A + B}$
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 15. Tabla de verdad de la función básica OR.

A partir de las funciones anteriores se derivan el resto de las funciones lógicas, ya que se forman con una combinación de ellas. Estas se describen a continuación:

- **XOR:** También denominada **OR exclusiva**, en esta función representada como $\boldsymbol{F = A \oplus B}$ o $\boldsymbol{F = \overline{A}B + A\overline{B}}$, el valor de la salida será igual a 1 si alguna de las dos entradas tiene un valor de 1; de lo contrario, cuando las dos entradas tienen un valor de 0 al mismo tiempo, entonces la salida será igual a 0. Esto se demuestra en la tabla de verdad de esta función, la cual se muestra a continuación en la Tabla 16:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{F = A \oplus B}$
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 16. Tabla de verdad de la función básica XOR.

- **NAND (|):** Esta función resulta de invertir la salida de una función AND. Se representa como $\boldsymbol{F = \overline{AB}}$ o $\boldsymbol{F = \overline{A} + \overline{B}}$ y el valor de la salida será igual a 0 si tanto la entrada A como la entrada B tienen un valor de 1; de lo contrario, la salida será igual a 0. Esto se demuestra en la tabla de verdad de esta función, la cual se muestra a continuación en la Tabla 17:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{F = \overline{A} + \overline{B}}$
0	0	1
0	1	1
1	0	1
1	1	0

Tabla 17. Tabla de verdad de la función básica NAND.

- **NOR (↓):** Esta función resulta de invertir la salida de una función OR. Se representa como $\boldsymbol{F = \overline{A + B}}$ o $\boldsymbol{F = \overline{A} \cdot \overline{B}}$ y el valor de la salida será igual a 1 si tanto la entrada A como la entrada B tienen un valor de 0; de lo contrario, la salida será igual a 0. Esto se demuestra en la tabla de verdad de esta función, la cual se muestra a continuación en la Tabla 18:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{F = \overline{A} \cdot \overline{B}}$
0	0	1
0	1	0
1	0	0
1	1	0

Tabla 18. Tabla de verdad de la función básica NOR.

- **XNOR:** También denominada **NOR exclusiva**, esta función resulta de invertir la salida de una función NOR. Se representa como $\boldsymbol{F = \overline{A \oplus B}}$ o $\boldsymbol{F = AB + \overline{A} \cdot \overline{B}}$ y el valor de la salida será igual a 1 si ambas entradas tienen un valor de 0 al mismo tiempo o si ambas tienen un valor de 0 al mismo tiempo. Esto se demuestra en la tabla de verdad de esta función, la cual se muestra a continuación en la Tabla 19:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{F = \overline{A \oplus B}}$
0	0	1
0	1	0
1	0	0
1	1	1

Tabla 19. Tabla de verdad de la función básica XNOR.

Implementación mediante conjuntos completos

Un **conjunto funcionalmente completo** se refiere a aquel conjunto de funciones que son **necesarias** y **suficientes**, de modo que de forma general todas las funciones lógicas pueden expresarse con las funciones AND, OR y NOT (Vingron, 2012). En este sentido, el conjunto {AND, OR, NOT} constituye un conjunto funcionalmente completo, ya que como describe (LaMeres, 2017), estos tres operadores pueden utilizarse para implementar las funciones NAND, NOR, XOR y XNOR. Adicionalmente, los conjuntos {AND, NOT} y {OR, NOT} también son conjuntos funcionalmente completos.

Como describe (LaMeres, 2017), todas las operaciones con AND y OR pueden ser reemplazadas con las funciones NAND y NOR, por las Leyes de DeMorgan. Por ello, los conjuntos {NAND} y {NOR} también son conjuntos funcionalmente completos.

- **Representación de la suma lógica con el conjunto {AND, NOT}**

La suma lógica se puede representar con una función que sólo contenga los operadores AND y NOT. En este sentido, se puede decir que:

$$A + B = (A' \cdot B')'$$

A continuación, se presenta la demostración:

$A + B = (A' \cdot B')'$

$A + B = ((A + B)')'$ $\longrightarrow$ Por ley de DeMorgan, donde $(A + B)' = A' \cdot B'$

$\boldsymbol{A + B = A + B}$ $\longrightarrow$ Por involución, es decir, $(A')' = A$

- **Representación del producto lógico con el conjunto {OR, NOT}**

El producto lógico se puede representar con una función que sólo contenga los operadores OR y NOT. En este sentido, se puede decir que:

$$A \cdot B = (A' + B')'$$

A continuación, se presenta la demostración:

$A \cdot B = (A' + B')'$

$A \cdot B = ((A \cdot B)')'$ $\longrightarrow$ Por ley de DeMorgan, donde $(A \cdot B)' = A' + B'$

$\boldsymbol{A \cdot B = A \cdot B}$ $\longrightarrow$ Por involución, es decir, $(A')' = A$

- **Representación de la suma lógica y el producto lógico con el conjunto {NAND}**

La suma lógica y el producto lógico pueden representarse con funciones que sólo contengan el operador NAND, el cual es denotado por el símbolo (|).

Antes de hacer la demostración, es necesario recordar que la función NAND es la negación de la función AND, de modo que:

$$A|B = (A \cdot B)'$$

donde al aplicar la ley de DeMorgan se obtiene que:

$$(A \cdot B)' = A' + B'$$

Recordando el teorema de idempotencia donde $A = A \cdot A$, se reemplaza como:

$$A' = (A \cdot A)'$$

donde al aplicar la definición de la función NAND, se obtiene que:

$$A' = A|A$$

Al representar la suma lógica con una función que sólo contenga el operador NAND, se puede decir que:

$$A + B = (A|A)|(B|B)$$

A continuación, se presenta la demostración:

$A + B = (A|A)|(B|B)$		
$A + B = A'|B'$	$\longrightarrow$	Por idempotencia, donde $A' = A|A$
$A + B = (A' \cdot B')'$	$\longrightarrow$	Por definición de la función NAND, donde: $A|B = (A \cdot B)'$
$A + B = ((A + B)')'$	$\longrightarrow$	Por ley de DeMorgan
$\boldsymbol{A + B = A + B}$	$\longrightarrow$	Por involución, es decir, $(A')' = A$

Al representar el producto lógico con una función que sólo contenga el operador NAND, se puede decir que:

$$A \cdot B = (A|B)|(A|B)$$

A continuación, se presenta la demostración:

$A \cdot B = (A|B)|(A|B)$

$A \cdot B = (A|B)'$ → Por idempotencia, donde $A' = A|A$

$A \cdot B = ((A \cdot B)')'$ → Por definición de la función NAND, donde:

$$A|B = (A \cdot B)'$$

$\boldsymbol{A \cdot B = A \cdot B}$ → Por involución, es decir, $(A')' = A$

- **Representación de la suma lógica y el producto lógico con el conjunto {NOR}**

La suma lógica y el producto lógico pueden representarse con funciones que sólo contengan el operador NOR, el cual es denotado por el símbolo (↓).

Antes de hacer la demostración, es necesario recordar que la función NOR es la negación de la función OR, de modo que:

$$A \downarrow B = (A + B)'$$

donde al aplicar la ley de DeMorgan se obtiene que:

$$(A + B)' = A' \cdot B'$$

Recordando el teorema de idempotencia donde $A = A + A$, se reemplaza como:

$$A' = (A + A)'$$

donde al aplicar la definición de la función NOR, se obtiene que:

$$A' = A \downarrow A$$

Al representar el producto lógico con una función que sólo contenga el operador NOR, se puede decir que:

$$A \cdot B = (A \downarrow A) \downarrow (B \downarrow B)$$

A continuación, se presenta la demostración:

$A \cdot B = (A \downarrow A) \downarrow (B \downarrow B)$

$A \cdot B = A' \downarrow B'$ ⟶ Por idempotencia, donde $A' = A \downarrow A$

$A \cdot B = (A' + B')'$ ⟶ Por definición de la función NOR, donde:

$$A \downarrow B = (A + B)'$$

$A \cdot B = ((A \cdot B)')'$ ⟶ Por ley de DeMorgan

$\boldsymbol{A \cdot B = A \cdot B}$ ⟶ Por involución, donde $(A')' = A$

Al representar la suma lógica con una función que sólo contenga el operador NOR, se puede decir que:

$$A + B = (A \downarrow B) \downarrow (A \downarrow B)$$

A continuación, se presenta la demostración:

$A + B = (A \downarrow B) \downarrow (A \downarrow B)$

$A + B = (A \downarrow B)'$ ⟶ Por idempotencia, donde $A' = A \downarrow A$

$A + B = ((A + B)')'$ ⟶ Por definición de la función NOR, donde:

$$A \downarrow B = (A + B)'$$

$\boldsymbol{A + B = A + B}$ ⟶ Por involución, donde $(A')' = A$

Capítulo IV: Puertas y funciones lógicas

En el Capítulo IV de este documento se definirá el concepto de compuerta lógica y se describirán los ocho tipos de compuertas básicas que existen, presentando sus correspondientes expresiones algebraicas, sus tablas de verdad y sus diagramas de temporización.

Posteriormente, se estudiarán dos métodos de simplificación de funciones lógicas: mediante los teoremas del álgebra booleana y mediante los mapas de Karnaugh. Este último método, permite simplificar funciones mediante un procedimiento sencillo, en el que no se requieren de los teoremas del álgebra de Boole.

Adicional a esto, se describirán más detalladamente las formas canónicas de una función lógica. Se recordará cada una de las formas canónicas y sus características principales y se estudiarán los conceptos de minterms y maxterms y sus características.

Puertas lógicas

Las **compuertas lógicas** son gráficos que simbolizan una función lógica (Vingron, 2012), y son los bloques básicos de cualquier circuito digital, ya que estos tienen la capacidad de tomar decisiones. Como explica (Kumar Sarkar et al., 2014), los circuitos lógicos realizan una serie de toma de decisiones para dar una respuesta lógica como resultado a un determinado problema o a un conjunto de condiciones de entrada. Todos los aparatos digitales, desde el dispositivo más simple hasta la computadora más sofisticada, están compuestos por compuertas conectadas en una amplia variedad de configuraciones.

Cada compuerta lógica puede tener una más variables o entradas a un operador lógico, con lo cual se obtiene una señal de salida (Jiménez Murillo, 2009). La salida de una compuerta consiste en situar su salida en 0 ó 1, dependiendo del estado de sus entradas y de la función lógica para la cual fue diseñada.

Debido a que las compuertas representan a las funciones lógicas básicas, cada compuerta lleva el mismo nombre de la función que representa. Como sucede con las funciones lógicas, el comportamiento de una compuerta puede describirse mediante:

- **Tabla de verdad:** Representa de forma ordenada todas las posibles combinaciones de estados lógicos que pueden existir en las entradas y el valor que toma la salida en cada caso.
- **Expresión algebraica:** Relaciona matemáticamente la salida con las entradas.
- **Diagrama de temporización:** Representa gráficamente el comportamiento de una compuerta con señales variables en el tiempo.

En electrónica digital, se pueden distinguir ocho compuertas lógicas. Basado en las descripciones de (Kumar Sarkar et al., 2014), a continuación se describirá cada una de estas compuertas:

a) Compuerta YES o Buffer

Se representa con la expresión algebraica $\boldsymbol{F = A}$ y su comportamiento es el siguiente:

- Si su entrada es cero, su salida será cero.
- Si su entrada es uno, su salida será uno.

A continuación, se presenta su correspondiente símbolo (Figura 19) y la descripción de su comportamiento mediante la tabla de verdad (Tabla 20) y el diagrama de temporización (Figura 20):

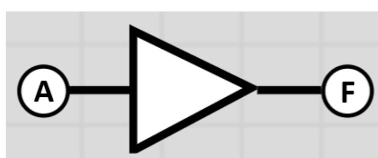


Figura 19. Símbolo de la compuerta YES. Fuente: Logicly.

$\boldsymbol{A}$	$\boldsymbol{A}$
0	0
1	1

Tabla 20. Tabla de verdad de la compuerta YES.

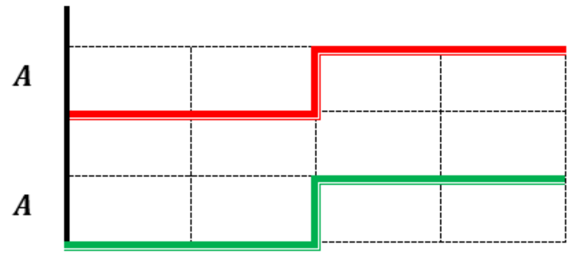


Figura 20. Diagrama de temporización de la compuerta YES. Elaboración propia.

b) Compuerta AND

Se representa con la expresión algebraica $\boldsymbol{F = A \cdot B}$ o $\boldsymbol{F = AB}$, y su comportamiento es el siguiente:

- Si todas sus entradas son uno, su salida será uno.
- Si al menos una de sus entradas es cero, su salida será cero.

A continuación, se presenta su correspondiente símbolo (Figura 21) y la descripción de su comportamiento mediante la tabla de verdad (Tabla 21) y el diagrama de temporización (Figura 22):

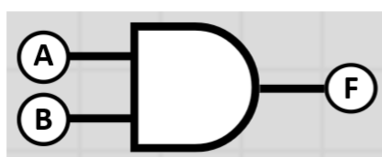


Figura 21. Símbolo de la compuerta AND. Fuente: Logicly.

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{F}$
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 21. Tabla de verdad de la compuerta AND.

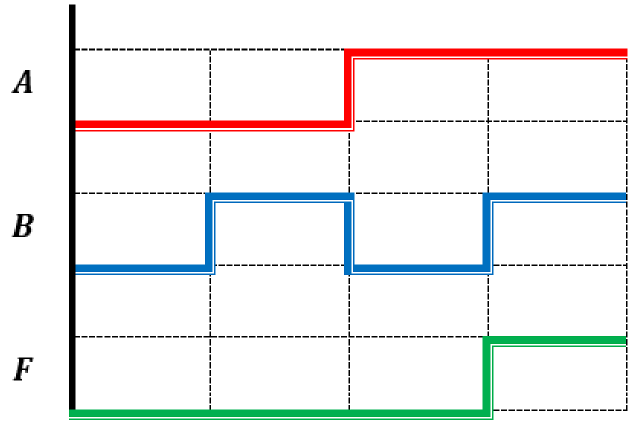


Figura 22. Diagrama de temporización de la compuerta AND. Elaboración propia.

c) Compuerta OR

Se representa con la expresión algebraica $\boldsymbol{F} = \boldsymbol{A} + \boldsymbol{B}$ y su comportamiento es el siguiente:

- Si al menos una de las entradas es uno, su salida será uno.
- Si todas las entradas son cero, su salida será cero.

A continuación, se presenta su correspondiente símbolo (Figura 23) y la descripción de su comportamiento mediante la tabla de verdad (Tabla 22) y el diagrama de temporización (Figura 24):

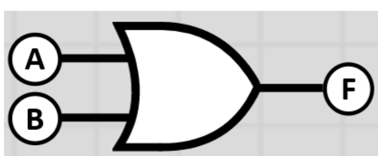


Figura 23. Símbolo de la compuerta OR. Fuente: Logicly.

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 22. Tabla de verdad de la compuerta OR.

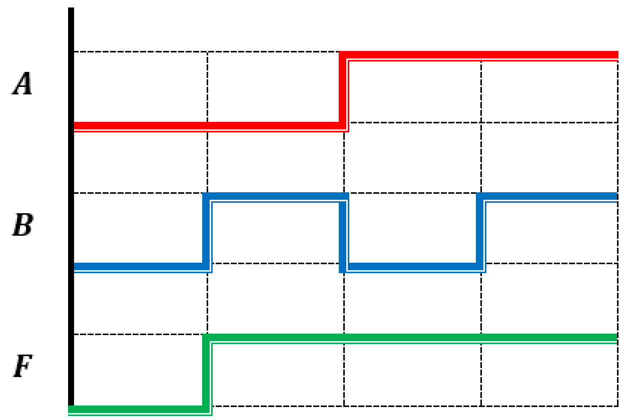


Figura 24. Diagrama de temporización de la compuerta OR. Elaboración propia.

d) Compuerta NOT

Se representa con la expresión algebraica $\boldsymbol{F} = \boldsymbol{A}'$ o $\boldsymbol{F} = \overline{\boldsymbol{A}}$, y su comportamiento es el siguiente:

- Si su entrada es cero, su salida será uno.
- Si su entrada es uno, su salida será cero.

A continuación, se presenta su correspondiente símbolo (Figura 25) y la descripción de su comportamiento mediante la tabla de verdad (Tabla 23) y el diagrama de temporización (Figura 26):

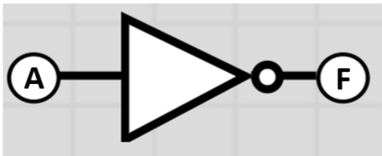


Figura 25. Símbolo de la compuerta NOT. Fuente: Logicly.

$\boldsymbol{A}$	$\boldsymbol{A}'$
0	1
1	0

Tabla 23. Tabla de verdad de la compuerta NOT.

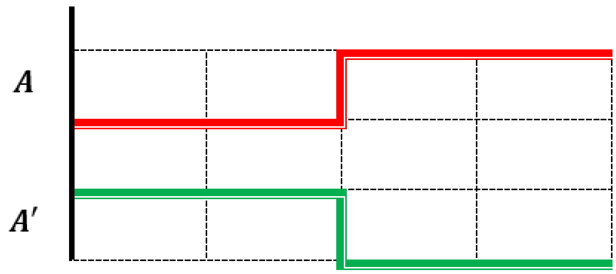


Figura 26. Diagrama de temporización de la compuerta NOT. Elaboración propia.

e) Compuerta NAND

Se representa con la expresión algebraica $\boldsymbol{F = \overline{AB}}$ o $\boldsymbol{F = \overline{A} + \overline{B}}$, y su comportamiento es el siguiente:

- Si al menos una de sus entradas es cero, su salida será uno.
- Si todas sus entradas son uno, su salida será cero.

A continuación, se presenta su correspondiente símbolo (Figura 27) y la descripción de su comportamiento mediante la tabla de verdad (Tabla 24) y el diagrama de temporización (Figura 28):

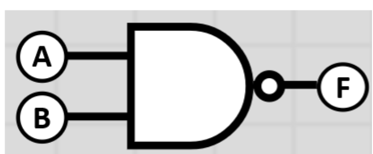


Figura 27. Símbolo de la compuerta NAND. Fuente: Logicly.

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Tabla 24. Tabla de verdad de la compuerta NAND.

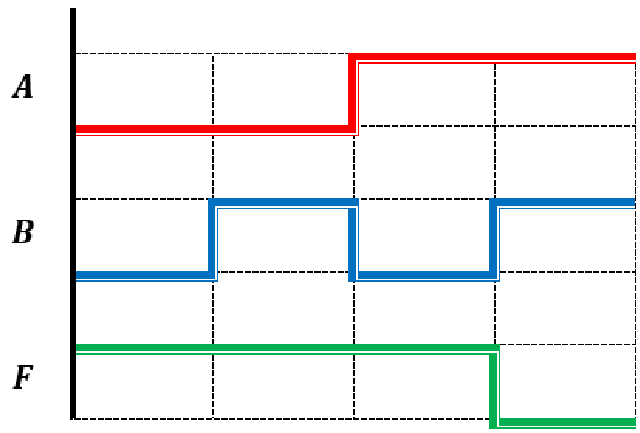


Figura 28. Diagrama de temporización de la compuerta NAND. Elaboración propia.

f) Compuerta NOR

Se representa con la expresión algebraica $\boldsymbol{F} = \overline{\boldsymbol{A} + \boldsymbol{B}}$ o $\boldsymbol{F} = \overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{B}}$, y su comportamiento es el siguiente:

- Si sus entradas son cero, su salida será uno.
- Si al menos de sus entradas son uno, su salida será cero.

A continuación, se presenta su correspondiente símbolo (Figura 29) y la descripción de su comportamiento mediante la tabla de verdad (Tabla 25) y el diagrama de temporización (Figura 30):

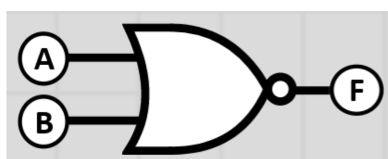


Figura 29. Símbolo de la compuerta NOR. Fuente: Logicly.

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Tabla 25. Tabla de verdad de la compuerta NOR.

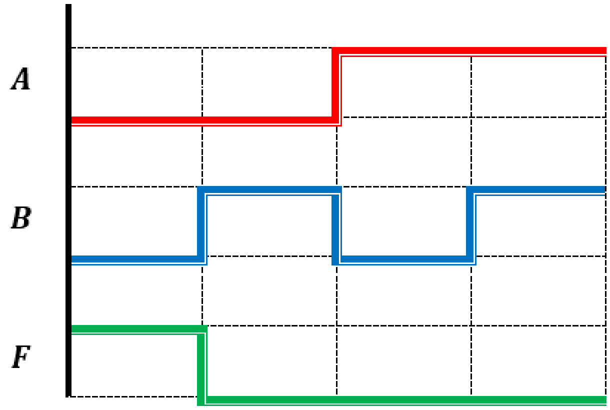


Figura 30. Diagrama de temporización de la compuerta NOR. Elaboración propia.

g) Compuerta XOR

Se representa con la expresión algebraica $\boldsymbol{F = A \oplus B}$ o $\boldsymbol{F = \overline{A}B + A\overline{B}}$, y su comportamiento es el siguiente:

- Si el número de entradas con valor igual a uno es impar, su salida será uno. De lo contrario, su salida será cero.

A continuación, se presenta su correspondiente símbolo (Figura 31) y la descripción de su comportamiento mediante la tabla de verdad (Tabla 26) y el diagrama de temporización (Figura 32):

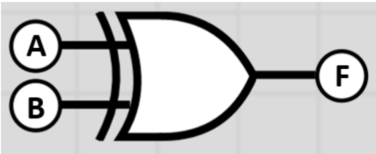


Figura 31. Símbolo de la compuerta XOR. Fuente: Logicly.

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 26. Tabla de verdad de la compuerta XOR.

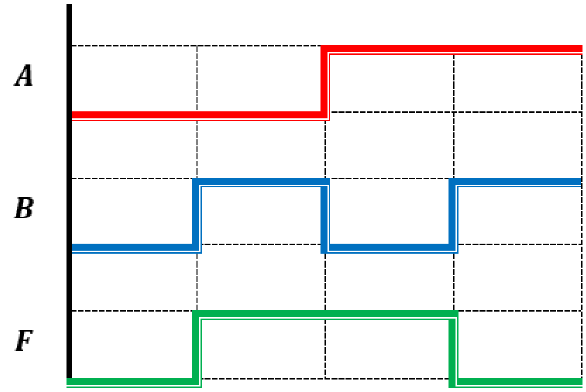


Figura 32. Diagrama de temporización de la compuerta XOR. Elaboración propia.

h) Compuerta XNOR

Se representa con la expresión algebraica $\boldsymbol{F = \overline{A \oplus B}}$ o $\boldsymbol{F = AB + \overline{A} \cdot \overline{B}}$, y su comportamiento es el siguiente:

- Si el número de entradas con valor igual a uno o cero es par, su salida será uno. De lo contrario, su salida será cero.

A continuación, se presenta su correspondiente símbolo (Figura 33) y la descripción de su comportamiento mediante la tabla de verdad (Tabla 27) y el diagrama de temporización (Figura 34):

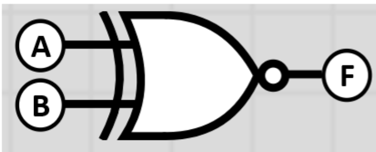


Figura 33. Símbolo de la compuerta XNOR. Fuente: Logicly.

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Tabla 27. Tabla de verdad de la compuerta XNOR.

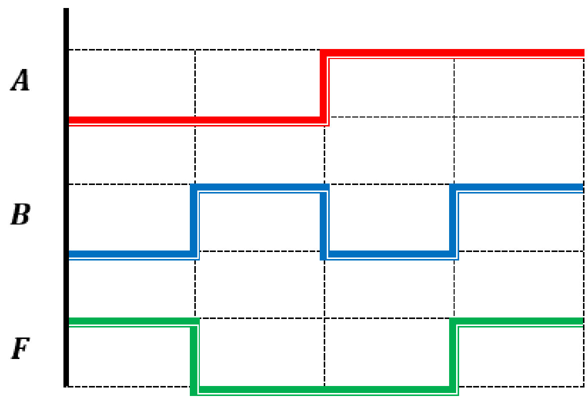


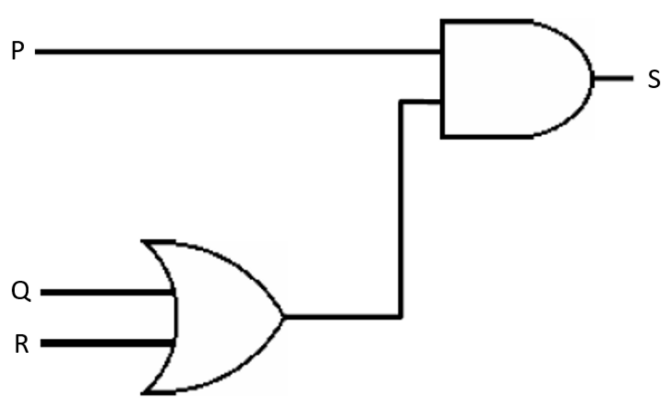
Figura 34. Diagrama de temporización de la compuerta XNOR. Elaboración propia.

A partir de una función lógica se pueden obtener un **diagrama lógico**, el cual es una representación en forma de símbolos de una función dada. El resultado de la función puede obtenerse con una tabla de verdad, ya que facilita la representación del comportamiento del circuito mostrando cada una de sus posibles combinaciones de entrada.

Ejemplo #1: Realizar el diagrama lógico de la siguiente función y obtener su tabla de verdad:

$$S = P \cdot (Q + R)$$

- **Solución:** Para su implementación, la función lógica dada requiere de una compuerta AND y una compuerta OR. El diagrama lógico quedaría como se muestra a continuación:



Para obtener la tabla de verdad, se obtiene el número de entradas para determinar la cantidad de combinaciones. La función lógica dada tiene tres variables, por lo que $n = 3$. Reemplazando el valor de n en la fórmula 2^n se obtiene que la tabla de verdad tendrá 8 posibles combinaciones. La tabla de verdad para la función $S = P \cdot (Q + R)$ se muestra a continuación:

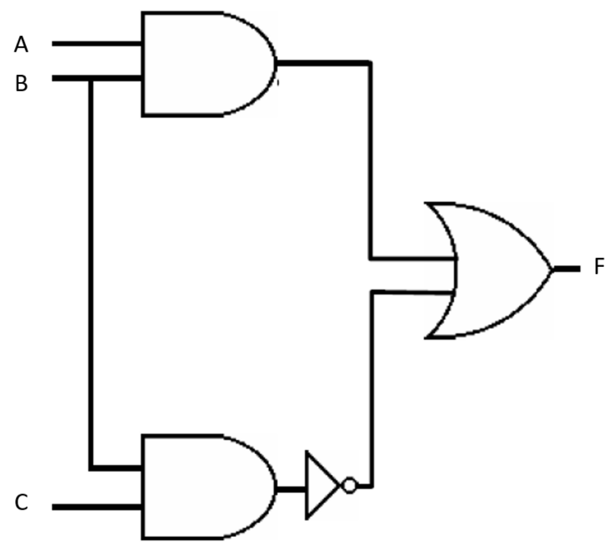
$\boldsymbol{P}$	$\boldsymbol{Q}$	$\boldsymbol{R}$	$\boldsymbol{Q+R}$	$\boldsymbol{S}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

La última columna de la tabla de verdad muestra el resultado de la salida de la función $S = P \cdot (Q + R)$, para cada combinación de entrada.

Ejemplo #2: Realizar el diagrama lógico de la siguiente función y obtener su tabla de verdad:

$$F = AB + (BC)'$$

- **Solución:** Para su implementación, la función lógica dada requiere de dos compuertas AND, una compuerta NOT y una compuerta OR. El diagrama lógico quedaría como se muestra a continuación:



Para obtener la tabla de verdad, se obtiene el número de entradas para determinar la cantidad de combinaciones. La función lógica dada tiene tres variables, por lo que $n = 3$. Reemplazando el valor de n en la fórmula 2^n se obtiene que la tabla de verdad tendrá 8 posibles combinaciones. La tabla de verdad para la función $F = AB + (BC)'$ se muestra a continuación:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{C}$	$\boldsymbol{AB}$	$\boldsymbol{(BC)'}$	$\boldsymbol{F}$
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	1	0	1

La última columna de la tabla de verdad muestra el resultado de la salida de la función $F = AB + (BC)'$, para cada combinación de entrada.

Simplificación de funciones lógicas

Como describe (Jiménez Murillo, 2009), muchas veces la función lógica obtenida a partir de un determinado problema no es óptima, por lo que es conveniente simplificar la expresión para facilitar la implementación de la función dada utilizando compuertas lógicas.

Para la simplificación de funciones lógicas se utilizan los teoremas y propiedades del álgebra booleana, de modo que se pueda obtener una función equivalente a la función que se está simplificando. A continuación, en la Tabla 28, se presenta un resumen de las propiedades y teoremas del álgebra booleana, basado en (Marcovitz, 2010):

Teoremas			
1a	$A + 0 = A$	**1b**	$A \cdot 1 = A$
2a	$A + 1 = A$	**2b**	$A \cdot 0 = 0$
3a	$A + A = A$	**3b**	$A \cdot A = A$
4a	$A + A' = 1$	**4b**	$A \cdot A' = 0$
5a	$A + B = B + A$	**5b**	$A \cdot B = B \cdot A$
6a	$A + (B + C) = (A + B) + C$	**6b**	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
7a	$(A + B)' = A' \cdot B'$	**7b**	$(A \cdot B)' = A' + B'$
8a	$A + (BC) = (A + B)(A + C)$	**8b**	$A(B + C) = (AB) + (AC)$
9a	$A + (A \cdot B) = A$	**9b**	$A \cdot (A + B) = A$
10a	$(A + B)(A + B') = A$	**10b**	$(A \cdot B) + (A \cdot B') = A$
11a	$A + (A' \cdot B) = A + B$	**11b**	$A \cdot (A' + B) = AB$
12a	$(A')' = A$		

Tabla 28. Resumen de los teoremas del álgebra booleana.

Ejemplo #1: Simplificar la siguiente función lógica:

$$S = PQR + PQR' + P'QR + P'QR'$$

- **Solución:** A continuación, se muestran los pasos y los teoremas que se pueden seguir para simplificar la función dada:

$S = PQR + PQR' + P'QR + P'QR'$

$S = PQ(R + R') + P'Q(R + R')$

$S = PQ(1) + P'Q(1)$ → por **4a**

$S = PQ + P'Q$

$S = Q(P + P')$

$S = Q(1)$ → por **4a**

$S = Q$ → por **1b**

Ejemplo #2: Simplificar la siguiente función lógica:

$$F = A' + B + (AC')(BC)$$

- **Solución:** Los pasos y los teoremas que se pueden seguir para simplificar la función dada son los siguientes:

$F = A' + B + (AC')(BC)$

$F = A' + B + AC'(BC)$ → por **6b**

$F = A' + B + AC'BC$ → por **6b**

$F = A' + B + ABCC'$ → por **5b**

$F = A' + B + AB(0)$ → por **4b**

$F = A' + B + 0$ → por **2b**

$F = A' + B$

Forma canónica de una función lógica

Anteriormente, se estudiaron las formas canónicas en las que puede estar una función lógica: la **suma de productos (SOP)** y el **producto de sumas (POS)**. Un resumen de las descripciones de cada una de las forma canónicas, se presenta a continuación en la Tabla 29:

	Suma de productos (SOP)	Producto de sumas (POS)
Descripción	Función en la que productos lógicos están conectados por operadores OR.	Función en la que sumas lógicas están conectadas por operadores AND.
Cómo obtenerla a partir de una tabla de verdad	Tomar las filas en las que la función es igual a 1 y las variables en esas filas cuyo valor es igual a 0, se colocan complementadas en la expresión SOP.	Tomar las filas en las que la función es igual a 0 y las variables en esas filas cuyo valor es igual a 1, se colocan complementadas en la expresión POS.

Tabla 29. Resumen de las formas canónicas de una función lógica.

Ejemplo: Para la tabla de verdad mostrada a continuación ¿Cuál sería la función expresada como POS? ¿Cuál sería la función expresada como SOP?

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- **Solución:** Considerando el procedimiento para obtener los términos de cada una de las formas canónicas, a continuación se muestra la función para esta tabla de verdad, expresada como POS y como SOP:

POS: $F = (A + B + C)(A + B + C')(A + B' + C)(A + B' + C')(A' + B + C)$

SOP: $F = AB'C + ABC' + ABC$

Ambas formas canónicas se componen de términos denominados **términos estándares**, como describe (Marcovitz, 2010). En cada término, cada una de las variables de entrada aparece exactamente una vez, ya sea complementada o sin complementar. Sin embargo, como explica (Katz & Borriello, 2005), ninguna variable aparece de ambas maneras en un solo término.

Ejemplo: Observe la siguiente función:

$$F = \bar{P}\bar{Q}R + \bar{P}Q\bar{R} + \bar{P}QR + P\bar{Q}\bar{R} + P\bar{Q}R$$

La función está expresada como SOP. Cada uno de los términos contiene todas las variables del problema una sola vez, siendo estas variables P, Q y R. Las variables aparecen complementadas o sin complementar, pero no aparecen de ambas formas en un mismo término.

Maxterms y minterms

En el caso de una suma de productos, el término estándar recibe el nombre de **minterm**; mientras que en el caso de un producto de sumas, el término estándar recibe el nombre de **maxterm**.

- **Minterm**

Un **minterm** se define como un **término producto** en el que cada variable del problema aparece exactamente una vez, complementada o no (Katz & Borriello, 2005). Se denota de la forma $\boldsymbol{m_i}$, donde $\boldsymbol{i}$ corresponde a la fila de la tabla de verdad en la que la función es exactamente igual a 1.

Es posible obtener una lista de minterms, en la que se indican los minterms de la tabla de verdad o diagrama lógico. Como describe (LaMeres, 2017), una lista de minterms se denota como $\boldsymbol{\Sigma m}$, seguido de los números de las filas en las que la función es igual a 1, colocados entre paréntesis. Otra forma de expresar la lista de minterms es simplemente colocando cada minterm conectado con operadores OR, similar a como sería una expresión SOP.

Ejemplo: A partir de la siguiente tabla de verdad, obtener la lista de minterms:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- **Solución:** Se toman aquellas filas en las que la función tiene un valor igual a 1. Para facilitar esto, se pueden numerar las filas de la tabla de verdad, como se muestra a continuación:

N°	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Para la tabla de verdad dada, la lista de minterms es la siguiente:

$$F = \Sigma m(2, 5, 6, 7)$$

La lista de minterms también puede expresarse de la siguiente forma:

$$F = m_2 + m_5 + m_6 + m_7$$

- **Maxterm**

Un **maxterm** se define como un **término suma** en el que cada variable del problema aparece exactamente una vez, complementada o no (Katz & Borriello, 2005). Se denota de la forma $\boldsymbol{M_i}$, donde $\boldsymbol{i}$ corresponde a la fila de la tabla de verdad en la que la función es exactamente igual a 0.

Es posible obtener una lista de maxterms, en la que se indican los maxterms de la tabla de verdad o diagrama lógico. Como describe (LaMeres, 2017), una lista de maxterms se denota como $\mathbf{\Pi M}$, seguido de los números de las filas en las que la función es igual a 0, colocados entre paréntesis. Otra forma de expresar la lista de minterms es simplemente colocando cada maxterm conectado con operadores AND, similar a como sería una expresión POS.

Ejemplo: A partir de la siguiente tabla de verdad, obtener la lista de maxterms:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- **Solución:** Se toman aquellas filas en las que la función tiene un valor igual a 0. Similar al ejemplo anterior, a continuación se muestra la tabla de verdad con las numeradas:

N°	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Para la tabla de verdad dada, la lista de maxterms es la siguiente:

$$F = \Pi\mathrm{M}(3, 4, 6, 7)$$

La lista de minterms también puede expresarse de la siguiente forma:

$$F = M_3 \cdot M_4 \cdot M_6 \cdot M_7$$

Como se mencionó anteriormente, cada término en una expresión SOP es un minterm, mientras que en una expresión POS cada término es un maxterm. En este sentido, una la lista de minterms o de maxterms se puede considerar una forma simplificada de una función expresada como SOP o POS, respectivamente.

Ejemplo: Obtener la lista de minterms y maxterms de la siguiente tabla de verdad. Después expresar la lista de minterms como una función SOP y la lista de maxterms como una función POS.

P	Q	R	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

- **Solución:** Numerando las filas de la tabla de verdad, como se muestra a continuación, se facilita la obtención de la lista de minterms y maxterms:

N°	P	Q	R	S
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

$$F = \Sigma m(1, 2, 3, 4, 5)$$

$$F = \Pi \mathrm{M}(0, 6, 7)$$

Para obtener la función expresada como SOP, se escriben las variables en su forma correspondiente, para cada fila que pertenece a la lista de minterms:

$m_1 = P'Q'R$

$m_2 = P'QR'$

$m_3 = P'QR$

$m_4 = PQ'R'$

$m_5 = PQ'R$

Se escriben los términos en su formato indicando las variables, para obtener la función expresada como SOP:

$$S = P'Q'R + P'QR' + P'QR + PQ'R' + PQ'R$$

Para obtener la función expresada como POS, se escriben las variables en su forma correspondiente, para cada fila que pertenece a la lista de maxterms:

$M_0 = PQR$

$M_6 = P'Q'R$

$M_7 = P'Q'R'$

Se escriben los términos en su formato indicando las variables, para obtener la función expresada como POS:

$$S = (P + Q + R)(P' + Q' + R)(P' + Q' + R')$$

Simplificación de funciones

Anteriormente, se estudió la simplificación de funciones lógicas mediante el uso de los teoremas del álgebra booleana. Sin embargo, como describe (Marcovitz, 2010), existen algunas desventajas, las cuales se mencionan a continuación:

- Algunas veces no se puede estar seguro de haber obtenido la mínima expresión de una función.
- Es posible que no se encuentre la mínima expresión de una función, aunque parezca que no se puede hacer nada más.
- El proceso de simplificación se complica cuando se tienen cuatro o más variables.
- Se requiere de habilidad para manipular las funciones, ya que no siempre se aplican los mismos teoremas o propiedades y el orden de aplicación de estos varía según la función.

Considerando esto, es conveniente describir otro método de simplificación de funciones lógicas, de modo que se puede escoger la técnica más adecuada.

En la siguiente sección de esta unidad se describirá otro método para la simplificación de funciones, denominado **diagramas de Karnaugh**.

Diagramas de Karnaugh

El método de **diagrama de Karnaugh** o **mapa de Karnaugh** fue propuesto inicialmente por Edward W. Veitch y luego modificado por Maurice Karnaugh (Jiménez Murillo, 2009). Un mapa de Karnaugh es un diagrama en forma de tabla, en el que cada celda contiene un minterm de la función. Como explica (Marcovitz, 2010), el mapa de Karnaugh permite encontrar de forma gráfica los términos más adecuados para simplificar una función expresada como suma de productos o como producto de sumas.

La cantidad de celdas de un mapa de Karnaugh dependen de la cantidad de variables en la función. Esto se calcula con la fórmula 2^n, donde n corresponde a la cantidad de variables.

Para construir un mapa de Karnaugh se colocan las variables de la función y las celdas se nombran siguiendo una secuencia binaria, pero asegurándose de que las celdas adyacentes sólo tengan un cambio en el valor de las variables, tanto en las filas como en las columnas:

- **Para 2 variables:** Se tiene que $n = 2$, por lo tanto $2^2 = 4\, celdas$. El mapa de Karnaugh quedaría de la siguiente manera, como se muestra en la Figura 35:

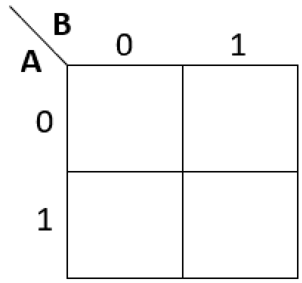


Figura 35. Representación de un mapa de Karnaugh de 2 variables. Elaboración propia.

- **Para 3 variables:** Se tiene que $n = 3$, por lo tanto $2^3 = 8\, celdas$. El mapa de Karnaugh quedaría de la siguiente manera, como se muestra en la Figura 36:

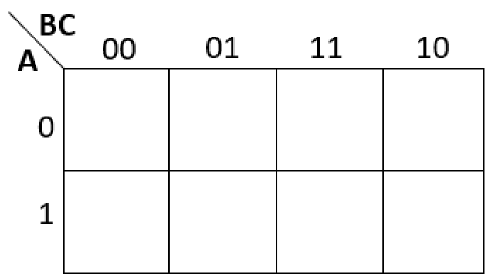


Figura 36. Representación de un mapa de Karnaugh de 3 variables. Elaboración propia.

- **Para 4 variables:** Se tiene que $n = 4$, por lo tanto $2^4 = 16\, celdas$. El mapa de Karnaugh quedaría de la siguiente manera, como se muestra en la Figura 37:

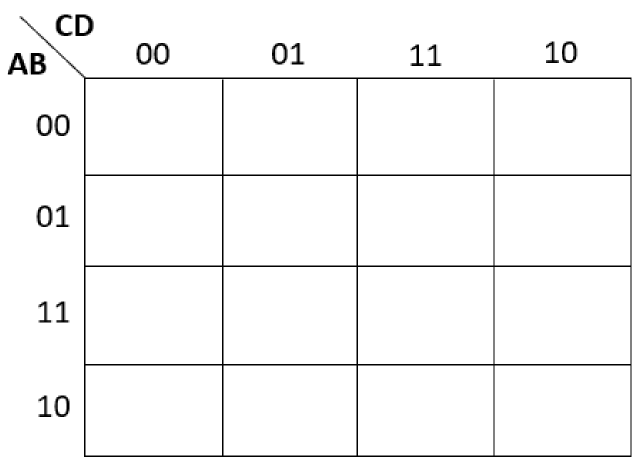


Figura 37. Representación de un mapa de Karnaugh de 4 variables. Elaboración propia.

Observe que al marcar las celdas con la secuencia binaria, se sigue el orden 00-01-11-10, en lugar del orden 00-01-10-11 como se haría en una tabla de verdad. Esto se debe a que al pasar de 01 a 11 sólo hay una variación en las variables, mientras que si se pasara de 01 a 10 habría dos variaciones. Esta regla aplica para la construcción de todos los mapas de Karnaugh.

Una vez se ha construido el mapa, se siguen los siguientes pasos, para obtener la simplificación de la función expresada como una suma de productos:

1. Colocar 1 en las celdas en las que la función tiene un valor de 1 y rellenar el resto con 0. Esto se hace dependiendo del valor de cada una de las variables, siguiendo las marcas colocadas durante la construcción del mapa.
2. Agrupar todos los 1 del mapa mediante rectángulos de $1, 2, 4, 8, \dots, 2^n$ elementos. Los grupos no deben contener 0. Al hacer la agrupación se debe tomar en cuenta lo siguiente:
    - Usar la menor cantidad de grupos posible, así como hacer los grupos tan grandes como sea posible.
    - Se puede agrupar los $1s$ de celdas adyacentes (**implicantes primos**), pero no se pueden hacer grupos diagonales. Aquel grupo que contenga un 1 que no pertenezca a ningún otro grupo, **debe** aparecer en el resultado final.
    - Puede haber coincidencias, es decir, que un 1 puede pertenecer a dos grupos.
    - Los grupos pueden rodear el mapa, es decir, se puede hacer un grupo con la celda más a la izquierda en una fila y la celda más a la derecha, o la celda superior en una columna con la celda inferior.
3. Cada grupo del mapa de Karnaugh representa un término producto (minterm). En los términos en los que una variable tiene un cambio de 0 a 1 o viceversa, dicha variable no se incluye en ese término. Las variables que no tienen cambios sí se incluyen en el término correspondiente. Finalmente, el resultado es una función simplificada de la función o de la tabla de verdad dada.

Ejemplo #1: A partir de la siguiente tabla de verdad, obtener la función simplificada utilizando el método del mapa de Karnaugh:

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

- **Solución:** Se construye el mapa de Karnaugh. Se tienen 3 variables por lo que $n = 3$, al reemplazar en la fórmula 2^n, se obtiene lo siguiente $2^3 = 8\ celdas$.

  Se rellenan con 1 aquellas celdas que correspondan a las combinaciones de variables en la tabla de verdad en las que la función es igual a 1. El resto de las celdas se rellenan con 0.

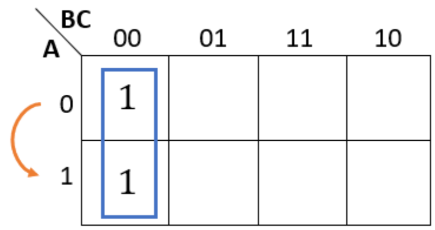
A \ BC	00	01	11	10
0	1	0	1	0
1	1	1	1	0

  Se agrupan los 1 que están en el mapa de Karnaugh:

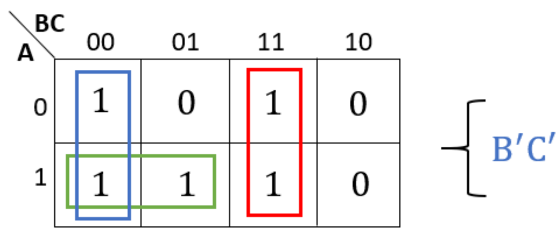
A \ BC	00	01	11	10
0	1	0	1	0
1	1	1	1	0

Se realiza la simplificación de los términos (grupos):

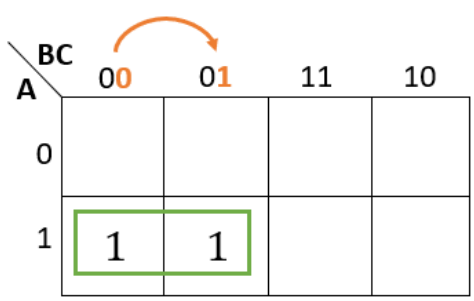
- Para el grupo indicado en color azul, la variable A pasa de 0 a 1 (como se indica con la flecha de color naranja en la siguiente figura) y por lo tanto, no se incluye en ese término de la función simplificada.



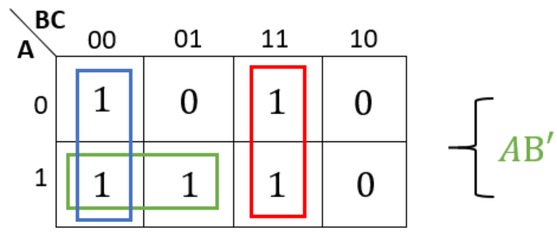
Los valores de las variables B y C no cambian y por lo tanto, sí se incluyen en ese término de la función. El término correspondiente a este grupo sería el siguiente:



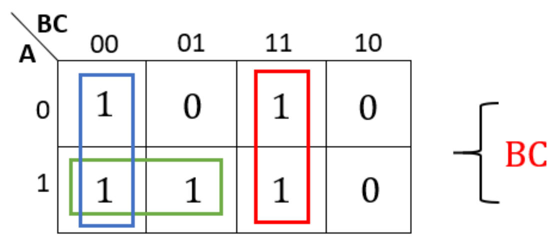
- Para el grupo indicado en color verde, la variable C pasa de 0 a 1 (como se indica con la flecha de color naranja en la siguiente figura) y por lo tanto, no se incluye en ese término de la función simplificada.



Los valores de las variables A y B no cambian y por lo tanto, sí se incluyen en ese término de la función. El término correspondiente a este grupo sería el siguiente:



- Para el grupo indicado en color rojo, nuevamente se observa que la variable A pasa de 0 a 1 y por lo tanto, no se incluye en ese término de la función simplificada. Los valores de las variables B y C no cambian y por lo tanto, sí se incluyen en ese término de la función. El término correspondiente a este grupo sería el siguiente:



Finalmente, se obtiene la función simplificada para la tabla de verdad dada. La función simplificada es la siguiente:

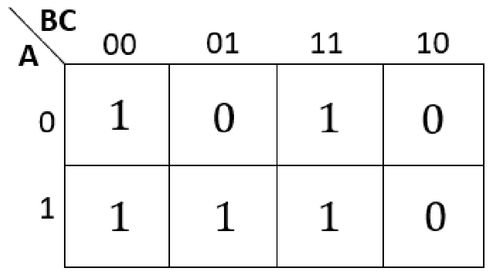
$$F = B'C' + AB' + BC$$

Ejemplo #2: Simplificar la siguiente función utilizando el método de mapa de Karnaugh:

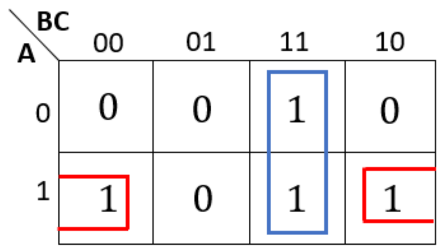
$$F = A'BC + AB'C' + ABC' + ABC$$

- **Solución:** Se construye el mapa de Karnaugh. Se tienen 3 variables por lo que $n = 3$, al reemplazar en la fórmula 2^n, se obtiene lo siguiente $2^3 = 8\ celdas$.

  A continuación, se colocan 1 en las combinaciones que se indican en los términos de la función dada. El mapa de Karnaugh quedaría de la siguiente manera:

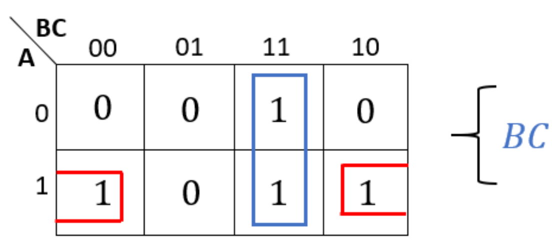


Se agrupan los 1 que están en el mapa de Karnaugh:

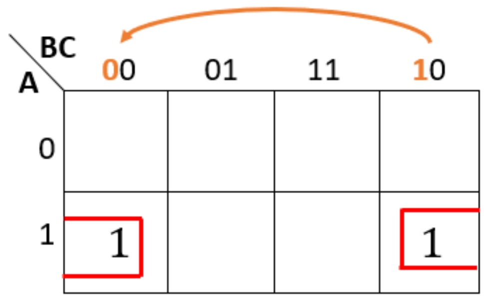


Se realiza la simplificación de los términos (grupos):

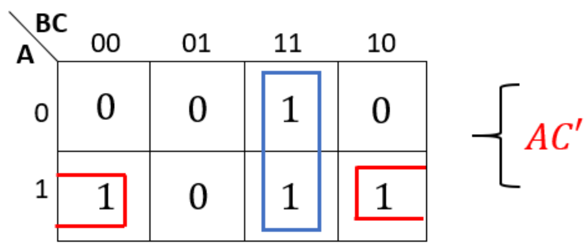
- Para el grupo indicado en color azul, la variable A pasa de 0 a 1 y por lo tanto, no se incluye en ese término de la función simplificada. Los valores de las variables B y C no cambian y por lo tanto, sí se incluyen en ese término de la función. El término correspondiente a este grupo sería el siguiente:



- Para el grupo indicado en color rojo, la variable B pasa de 1 a 0 (como se indica con la flecha de color naranja en la siguiente figura) y por lo tanto, no se incluye en ese término de la función simplificada.



Los valores de las variables A y C no cambian y por lo tanto, sí se incluyen en ese término de la función. El término correspondiente a este grupo sería el siguiente:



Finalmente, se obtiene una función simplificada de la función original dada en el problema:

$$F = BC + AC'$$

Ejemplo #3: Simplificar la siguiente función utilizando el método de mapa de Karnaugh:

$$F = A'BCD + ABC'D' + ABCD' + ABCD$$

- **Solución:** Se construye el mapa de Karnaugh. Se tienen 4 variables por lo que $n = 4$, al reemplazar en la fórmula 2^n, se obtiene lo siguiente $2^4 = 16\ celdas$. El mapa de Karnaugh quedaría de la siguiente manera:

  A continuación, se colocan 1 en las combinaciones que se indican en los términos de la función dada. El mapa de Karnaugh quedaría de la siguiente manera:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	0	1	1
10	0	0	0	0

Se agrupan los 1 que están en el mapa de Karnaugh:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	0	1	1
10	0	0	0	0

Se realiza la simplificación de los términos (grupos). Para este problema se obtienen los siguientes términos:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	0	1	1
10	0	0	0	0

BCD

ABD'

Observe que para el grupo indicado en color azul, la variable A tiene un cambio de valor de 0 a 1. Sucede algo similar con el término indicado en color rojo, ya que la variable C tiene un cambio de valor de 1 a 0.

Finalmente, se obtiene la función simplificada de la función original:

$$F = ABD' + BCD$$

Bibliografía

Conradie, W., & Goranko, V. (2015). *Logic and Discrete Mathematics: A Concise Introduction*. Wiley.

Contreras Oré, F. A. (2017). La axiomática. *Horizonte de La Ciencia*, *7*(12), 111–121.

Del Callejo-Canal, D. D., Del Callejo-Canal, E., & Canal-Martínez, M. E. (2016). *Lógica: El pensamiento matemático*. Universidad Veracruzana.

Epp, S. (2011). *Discrete Mathematics with Applications* (4th ed.). Cengage Learning.

Garrido, M. (2005). *Lógica simbólica* (4th ed.). Editorial Tecnos.

Jenkyns, T., & Stephenson, B. (2018). *Fundamentals of Discrete Math for Computer Science: A Problem-Solving Primer* (2nd ed.). Springer.

Jiménez Murillo, J. A. (2009). *Matemáticas para la Computación* (1era ed.). Alfaomega.

Katz, R. H., & Borriello, G. (2005). *Contemporary Logic Design* (2nd ed.). Pearson Prentice Hall.

Kumar Sarkar, S., Kumar De, A., & Sarkar, S. (2014). *Foundations of Digital Electronics and Logic Design*. Pan Stanford Publishing.

LaMeres, B. J. (2017). *Introduction to Logic Circuits & Logic Design with VHDL* (1st ed.). Springer.

Levin, O. (2018). *Discrete Mathematics: An Open Introduction* (3rd ed.). Oscar Levin.

Lipschutz, S., & Lipson, M. (2009). *Matemáticas discretas* (3rd ed.). McGraw Hill Educación.

Marcovitz, A. B. (2010). *Introduction to Logic Design* (3rd ed.). McGraw-Hill.

O'Regan, G. (2013). *Mathematics in Computing*. Springer.

Stein, C., Drysdale, R. L., & Bogart, K. (2011). *Discrete Mathematics for Computer Scientists*. Addison-Wesley.

Vingron, S. P. (2012). *Logic Circuit Design: Selected Methods*. Springer.

Anexos 1: Pruebas Rápidas

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #1

Nombre: ______________________ Cédula: ________________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. Seleccione los tipos de lógica más importantes.
   - **a.** Lógica proposicional
   - **b.** Lógica informal
   - **c.** Lógica difusa
   - **d.** Lógica simbólica
   - **e.** Lógica de predicados

2. Tanto la lógica de predicados como la proposicional, consisten en las siguientes partes: Sintaxis, Semántica y Prueba (proof). Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

3. La lógica de predicado es una lógica en el nivel de la sentencia, en el que la unidad más pequeña es una oración. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

4. La lógica proposicional hace frente a las deficiencias de la lógica proposicional introduciendo dos nuevas elementos: predicados y cuantificadores. Seleccione una:
   - **a.** Verdadero

b. Falso

5. La Prueba (proof) es el principio del cálculo, los axiomas y de las reglas de inferencia. Seleccione una:

a. Verdadero

b. Falso

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #2

Nombre: ______________________ Cédula: ________________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. Un sistema axiomático, también denominado sistema formal o cálculo axiomático, es una lista de conceptos y hechos básicos a partir de los cuales se derivan otros conceptos y teoremas, a través de la definición y la deducción. Seleccione una:
   a. Verdadero
   b. Falso

2. Un sistema axiomático no es más que un mecanismo capaz de generar una teoría, es decir, afirmaciones para un determinado dominio de problemas. Seleccione una:
   a. Verdadero
   b. Falso

3. Todo sistema axiomático se compone de los siguientes elementos que se muestran a continuación:
   i. Permiten generar nuevas fórmulas a partir de una o más fórmulas dadas:
      a. Axiomas
      b. Reglas de inferencia
      c. Lenguaje
   ii. Son la fórmulas del sistema:
      a. Axiomas
      b. Reglas de inferencia

c. Lenguaje

iii. Se compone de una tabla de símbolos (alfabeto) y de reglas para formar fórmulas (gramática):

a. Axiomas

b. Reglas de inferencia

c. Lenguaje

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #3

Nombre: ______________________ Cédula: ________________ Grupo: ________

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Llenar los espacios. Coloque los conceptos dados en la posición correcta.

Axiomas	Reglas de inferencia	Teorema

Al demostrar una teoría, lo que se hace es validar sus proposiciones, a partir de los ________________________________ o de las fórmulas obtenidas como consecuencia de las ________________________________. Al final de este proceso se generará un ________________________________, que es una proposición que puede ser demostrada a través de los axiomas.

II Parte. Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. Los axiomas de un sistema axiomático pueden tener las siguientes propiedades.
   - **i.** Si no hay contradicciones entre ellos; es decir, si se deduce una proposición, a la vez no se puede deducir su negación.
     - **a.** Completos
     - **b.** Independiente
     - **c.** Consistentes
   - **ii.** Si cada afirmación puede ser probada por axiomas.
     - **a.** Completos
     - **b.** Independiente
     - **c.** Consistentes
   - **iii.** Si no se puede deducir a partir de los otros axiomas.
     - **a.** Completos
     - **b.** Independiente

    c. Consistentes

2. Los axiomas se construyen utilizando concatenaciones lógicas. De esta manera, las proposiciones tienen valores de verdad bien definidos, el cual puede ser “verdadero” o “falso”. Seleccione una:
    a. Verdadero
    b. Falso

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #4

Nombre: ______________________ Cédula: ________________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. Una proposición condicional es aquella en la que se conectan dos enunciados y la verdad del segundo enunciado q, está condicionada por la verdad del primer enunciado p. Seleccione una:
   a. Verdadero
   b. Falso

2. Una proposición condicional se denota utilizando el operador condicional →, de la siguiente manera: $p \rightarrow q$. Seleccione una:
   a. Verdadero
   b. Falso

II Parte. Llenar los espacios. Coloque los conceptos dados en la posición correcta.

1. En la proposición condicional $p \rightarrow q$ se pueden distinguir dos partes:
   - La ________________________________, que es la parte antes del operador, es decir p.
   - La ________________________________, que es la parte después del operador, es decir $\boldsymbol{q}$.

Hipótesis o antecedente	Conclusión o consecuente

2. La proposición condicional $p \rightarrow q$ puede leerse como "si p ________________________________ q", "p ________________________________ q" o "p sólo si q".

Entonces
Implica

3. Existen algunas variaciones de la proposición condicional, las cuales se definen a continuación:

a. ________________________________: Para una proposición condicional $p \rightarrow q$, el inverso es $\neg p \rightarrow \neg q$. Es decir, es la negación tanto del antecedente como del consecuente de la proposición original.

b. ________________________________: Para una proposición condicional $p \rightarrow q$, el recíproco es $q \rightarrow p$. Es decir, el consecuente pasa a ser el antecedente y el antecedente pasa a ser el consecuente.

c. ________________________________: Para un proposición $p \rightarrow q$, el contrapositivo es $\neg q \rightarrow \neg p$. Es decir, es la negación del antecedente y del consecuente del inverso de la proposición original.

Inverso
Recíproco
Contrapositivo

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #5

Nombre: ______________________ Cédula: ________________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. Un cuantificador convierte una declaración simbólica acerca de cualquier elemento del universo en una declaración acerca del universo. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

2. Recordemos que un sistema axiomático se compone de un lenguaje, de axiomas y de reglas de inferencia, y que a partir de los axiomas se derivan otros razonamientos y proposiciones. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

3. El cálculo proposicional, también denominado lógica difusa. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

II Parte. Llenar los espacios. Coloque los conceptos dados en la posición correcta.

Se pueden distinguir dos tipos de cuantificadores, los cuales indican que el predicado es verdadero o falso, ya sea para todos los posibles objetos o para algunos objetos del universo. Estos dos cuantificadores se describen a continuación:

a) ________________________________: Afirma que una declaración acerca de una variable es verdadera para todos los valores de la variable en el universo. El símbolo utilizado para denotar este cuantificador es ________________ y se lee "para cada" o "para todo".

b) ________________________________: Afirma que cierto elemento del universo existe. El símbolo utilizado para denotar este cuantificador es ________________ y se lee "existe" o "hay algunos".

Cuantificador universal	$\forall$
Cuantificador existencial	$\exists$

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #6

Nombre: ______________________ Cédula: _______________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. El álgebra booleana, también llamada álgebra de Boole, en informática y matemática, es una estructura algebraica que esquematiza las operaciones lógicas. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

2. Las proposiciones lógicas booleanas no puedan manipularse de manera algebraica. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso
3. A continuación se describen las propiedades de las operaciones del álgebra booleana: Existencia de neutros

   • $A + 0 = A$

   • $A \cdot 1 = A$

   Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

II Parte. Llenar los espacios. Coloque los conceptos dados en la posición correcta.

1. Dentro de la lógica booleana existen una serie de definiciones básicas:

a) ______________________________: Símbolo que puede ser sustituido por un elemento del conjunto B = {0, 1}.

b) ______________________________: Es un valor que pertenece al conjunto {0, 1}.

c) ______________________________: Se compone de variables, constantes y operadores.

d) ______________________________: Una función booleana de n variables $f(x1, x2, \ldots, xn)$, es una expresión que mapea f a un valor del conjunto booleano B.

e) ______________________________: Es una variable o su complemento.

Variable	Constante	Expresión	Función	Literal

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #7

Nombre: ______________________ Cédula: _______________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. Una función lógica, también denominada función booleana, describe las operaciones necesarias para que determinadas entradas de un circuito produzcan una salida que sólo puede tomar los valores de 0 y 1. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

2. Una tabla de verdad es una especificación formal que describe el comportamiento exacto de un circuito para todas las posibles entradas, especificando las salidas para cada combinación de entradas. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

3. Una tabla de verdad tendrá $\mathbf{2}$ elevado a la $\boldsymbol{n}$ filas, donde n es la cantidad de funciones; mientras que la cantidad columnas será una más que la cantidad de funciones al cuadrado formando así una función lógica. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

II Parte. Llenar los espacios. Coloque los conceptos dados en la posición correcta.

1. El álgebra booleana consiste en un conjunto B, que contiene:
   - Dos elementos → 0 (falso) y 1 (verdadero)

- Dos operaciones binarias → ________________ o producto lógico (·) y ________________ o suma lógica (+).
- Una operación unaria → ________________ o negación, representado como ________________ o (′) . También se denomina complemento.

AND	OR	NOT	$(-)$

2. Una forma canónica es un término estándar con el cual se representa una función y permite comparar funciones booleanas que están expresadas en términos algebraicos. Se distinguen dos formas canónicas:

- ________________________________: Es aquella expresión en la que una o más productos lógicos de uno o más literales, están conectados por operadores ________________.
- ________________________________: Es aquella expresión en la que una o más sumas lógicas de uno o más literales, están conectados por operadores ________________.

Suma de productos (SOP)	OR
Producto de sumas (POS)	AND

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #8

Nombre: ______________________ Cédula: ________________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. Las funciones lógicas complejas se componen de una combinación de las funciones lógicas básicas, que son las mismas operaciones básicas del álgebra booleana. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

II Parte. Complete las tablas de verdad dadas a continuación.

1. XNOR: También denominada NOR exclusiva, esta función resulta de invertir la salida de una función NOR. Se representa como $\boldsymbol{F = \overline{A \oplus B}}$ o $\boldsymbol{F = AB + \overline{A} \cdot \overline{B}}$ y el valor de la salida será igual a 1 si ambas entradas tienen un valor de 0 al mismo tiempo o si ambas tienen un valor de 0 al mismo tiempo.

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{F = \overline{A \oplus B}}$
0	0	
0	1	
1	0	
1	1	

2. NOR (↓): Esta función resulta de invertir la salida de una función OR. Se representa como $\boldsymbol{F = \overline{A + B}}$ o $\boldsymbol{F = \overline{A} \cdot \overline{B}}$ y el valor de la salida será igual a 1 si tanto

la entrada A como la entrada B tienen un valor de 0; de lo contrario, la salida será igual a 0.

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{F = \overline{A} \cdot \overline{B}}$
0	0	
0	1	
1	0	
1	1	

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #9

Nombre: ______________________ Cédula: ________________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

Llenar los espacios. Coloque los conceptos dados en la posición correcta.

1. Las ________________________________ son gráficos que simbolizan una ________________________________, y son los bloques básicos de cualquier ________________________________, ya que estos tienen la capacidad de tomar decisiones.

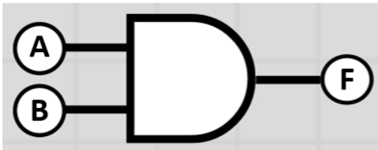
Compuertas lógicas	Función lógica	Circuito digital

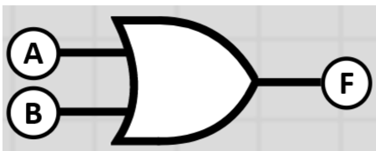
2. Debido a que las compuertas representan a las funciones lógicas básicas, cada compuerta lleva el mismo nombre de la función que representa. Como sucede con las funciones lógicas, el comportamiento de una compuerta puede describirse mediante:

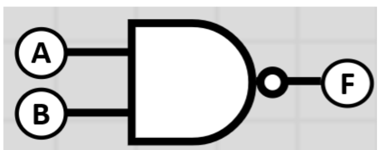
- ________________________________: Representa de forma ordenada todas las posibles combinaciones de estados lógicos que pueden existir en las entradas y el valor que toma la salida en cada caso.
- ________________________________: Relaciona matemáticamente la salida con las entradas.
- ________________________________: Representa gráficamente el comportamiento de una compuerta con señales variables en el tiempo.

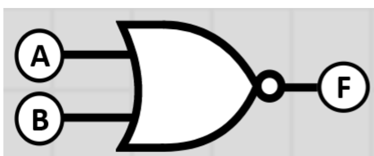
Tabla de verdad	Expresión algebraica	Diagrama de temporización

3. Cada compuerta lógica puede tener una más o variables o entradas a un operador lógico, con lo cual se obtiene una señal de salida. La salida de una compuerta consiste en situar su salida en 0 ó 1, dependiendo del estado de sus entradas y de la función lógica para la cual fue diseñada. Relacione cada símbolo de la compuerta con su respectivo nombre.









AND	*OR*	*NAND*	*NOR*

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Asignación #10

Nombre: ______________________ Cédula: ________________ Grupo: ________

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. Las formas canónicas en las que puede estar una función lógica: la suma de productos (SOP) y el producto de sumas (POS). Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

2. Un maxterm se define como un término suma en el que cada variable del problema aparece exactamente una vez, complementada o no. Seleccione una:
   - **a.** Verdadero
   - **b.** Falso

3. Un minterm se define se define como un término producto en el que cada variable del problema aparece exactamente una vez, complementada o no. Seleccione una:
   - **c.** Verdadero
   - **d.** Falso

II Parte. Llenar los espacios. Coloque los conceptos dados en la posición correcta.

Suma de productos (SOP)	productos lógicos
Producto de sumas (POS)	sumas lógicas

Forma canónica de una función lógica. ________________________________ es la función en la que ________________________________ están conectados por operadores OR. ________________________________ es la función en la que ________________________________ están conectadas por operadores AND.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #1

Nombre: ______________________ Cédula: ________________ Grupo: ________

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Cierto (C) y falso (F).

1. ____ Las constantes son frases que cuantifican objetos en el discurso.
2. ____ La lógica es una rama práctica de la filosofía.
3. ____ La cardinalidad de un conjunto es la cantidad de elementos que contiene un conjunto.
4. ____ En la lógica proposicional, la unidad más pequeña es una oración.
5. ____ El conjunto potencia de un conjunto A son todos los subconjuntos de A.
6. ____ Los elementos básicos de la lógica proposicional son las proposiciones y las conectivas.
7. ____ Las principales ramas de la lógica son la lógica formal y la lógica informal.
8. ____ El cuantificador universal se denota por el símbolo $\exists$.
9. ____ La teoría de conjuntos fue propuesta por George Boole.
10. ____ Un predicado es una plantilla de frase verbal que describe una propiedad de objetos representados por variables.

II Parte. Desarrollo.

1. ¿Cuál es la diferencia entre la lógica proposicional y la lógica de predicado?

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #2

Nombre: ______________________ Cédula: _______________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. A la colección de objetos considerada como un todo se le denomina: _______________________________.

2. Mencione tres tipos de conjuntos especiales: _______________________________, _______________________________ y _______________________________.

3. El cuantificador universal es denotado por el símbolo: _______________.

4. El cuantificador existencial es denotado por el símbolo: _______________.

5. La lógica _______________________________ estudia el razonamiento, la inferencia y la argumentación del lenguaje natural, mientras que la lógica _______________________________ es el estudio abstracto de las proposiciones, argumentos y afirmaciones, a través de un conjunto de reglas.

6. Nombre que reciben los objetos de un conjunto: _______________________________.

II Parte. Desarrollo.

1. Obtenga el producto cartesiano de los conjuntos $A = \{1, 3, 5\}$ y $B = \{2, 4\}$.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #3

Nombre: ______________________ Cédula: ________________ Grupo: ________

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Cierto (C) y falso (F).

1. ____ El cuantificador existencial afirma que una declaración acerca de una variable es verdadera para todos los valores de la variable en el universo.
2. ____ Un axioma es independiente si no se puede deducir a partir de otros axiomas.
3. ____ Un sistema axiomático también puede denominarse sistema formal.
4. ____ La axiomatización formal usa una lengua natural formalizada.
5. ____ El contrapositivo de la proposición $p \to q$ es $\neg q \to \neg p$.
6. ____ Las tautologías no pueden utilizarse en la demostración de teoremas.
7. ____ Una proposición condicional es lógicamente equivalente a su recíproco.
8. ____ El inverso de la proposición $p \to q$ es $q \to p$.
9. ____ El cálculo proposicional estudia el álgebra de las proposiciones.
10. ____ Todo sistema axiomático se compone de lenguaje, axiomas y reglas de inferencia.

II Parte. Desarrollo.

1. Describa las siguientes propiedades de los axiomas en un sistema axiomático:

   a. Consistente
   b. Completo

2. ¿Qué es un sistema axiomático?

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #4

Nombre: ______________________ Cédula: ________________ Grupo: ________

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Los axiomas son _______________________________ si cada afirmación puede ser probada por axiomas.

2. Nombre de la regla de inferencia que indica que si una primera afirmación implica una segunda afirmación y la primera afirmación es verdadera, entonces la segunda afirmación es verdadera: _______________________________.

3. Nombre de la regla de inferencia que indica que si una primera afirmación implica una segunda afirmación y la segunda afirmación no es verdadera, entonces la primera afirmación no es verdadera: _______________________________.

4. Todo sistema axiomático se compone de los siguientes elementos: _______________________________, _______________________________ y _______________________________.

5. Forma lógica que consiste en una función que toma premisas y analiza su sintaxis para devolver una conclusión: _______________________________.

6. Indique el símbolo utilizado para representar los siguientes operadores lógicos:

a. Disyunción: ________________ b. Conjunción: ________________

II Parte. Desarrollo.

1. ¿Cuál es la diferencia entre una proposición condicional y una proposición bicondicional?

2. Dada la siguiente proposición: *“Si mañana no llueve, entonces caminaré para llegar a la universidad”*, indique cuál es la hipótesis y cuál es la conclusión.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #5

Nombre: ______________________ Cédula: ________________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto   Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Todo lenguaje se compone de dos elementos: ________________________________ y ________________________________.

2. Dos tipos de relaciones básicas de conjuntos son: ________________________________ y ________________________________.

3. Símbolo que representa al conjunto vacío: ________________.

4. Una ________________________________ es la correspondencia entre dos elementos de dos conjuntos.

5. Nombre del gráfico que representa la relación entre los elementos de un conjunto, con regiones encerradas dentro de un plano: ________________________________.

II Parte. Desarrollo.

1. Obtenga el conjunto potencia del siguiente conjunto: $B = \{a, b\}$.

2. Obtenga el recíproco y el inverso de la siguiente proposición condicional: *“Si el polígono tiene cuatro lados, entonces es un rectángulo”.*

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #6

Nombre: ______________________ Cédula: ________________ Grupo: ________

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Cierto (C) y falso (F).

1. ____ La suma de productos también se denomina forma canónica conjuntiva.
2. ____ Una función booleana también se denomina función lógica.
3. ____ Para representar el comportamiento de un circuito de 3 entradas en una tabla de verdad, dicha tabla debe tener 10 filas o combinaciones.
4. ____ Dos formas de representar a las funciones lógicas son los diagramas de compuertas y los mapas de Karnaugh.
5. ____ Un conjunto funcionalmente completo es aquel conjunto que puede expresarse solamente con las funciones AND y OR.
6. ____ Las funciones AND y OR solamente pueden tener dos entradas.
7. ____ A la estructura algebraica que esquematiza las operaciones lógicas se le denomina álgebra de Boole.
8. ____ Un teorema fundamental del álgebra booleana es la dualidad.
9. ____ Las señales pueden ser analógicas o digitales.
10. ____ Un producto de sumas es una expresión en la que una o más sumas lógicas se conectan por operadores AND.

II Parte. Desarrollo.

1. Mencione las cinco definiciones básicas que forman parte de la lógica booleana.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #7

Nombre: ______________________ Cédula: ________________ Grupo: ________

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Especificación formal que describe el comportamiento de un circuito especificando las salidas para cada combinación de entradas: _______________________________.

2. Nombre de la expresión en la que una o más sumas lógicas de uno o más literales, están conectados por operadores AND: _______________________________.

3. Nombre de la expresión en la que uno o más productos lógicos de uno o más literales, están conectados por operadores OR: _______________________________.

4. Las operaciones binarias que forman parte del álgebra booleana son: _______________________________ y _______________________________.

5. Mencione dos formas de representar una función lógica: _______________________________ y _______________________________.

II Parte. Desarrollo.

1. Obtenga la tabla de verdad de la siguiente función: $F = \bar{A}\bar{B}C + A\bar{B}C + ABC$.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #8

Nombre: ______________________ Cédula: ________________ Grupo: ________

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Cierto (C) y falso (F).

1. ____ Un maxterm se representa como m_i.
2. ____ En un producto de sumas, el término estándar se denomina maxterm.
3. ____ Dos formas de representar el comportamiento de una función lógica son las tablas de verdad y los diagramas de temporización.
4. ____ Una lista de minterms se representa como Σm.
5. ____ En electrónica digital, se distinguen cuatro compuertas lógicas.

II Parte. Desarrollo.

1. Describa qué es:

a. Suma de productos
b. Producto de sumas

2. Dibuje el símbolo de las siguientes compuertas lógicas:

a. OR
b. NAND
c. NOT
d. XOR
e. AND
f. NOR

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #9

Nombre: ______________________ Cédula: ________________ Grupo: ________

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Las siguientes son formas de representar el comportamiento de una compuerta lógica:

   a. ________________________________: Representa gráficamente el comportamiento de una compuerta con señales variables en el tiempo.
   b. ________________________________: Representa de forma ordenada todas las posibles combinaciones de estados lógicos que pueden existir en las entradas y el valor que toma la salida en cada caso.
   c. ________________________________: Relaciona matemáticamente la salida con las entradas.

2. Un ________________________________ es un término producto en el que cada variable del problema aparece exactamente una vez, complementada o no.

3. Un ________________________________ es un término suma en el que cada variable del problema aparece exactamente una vez, complementada o no.

II Parte. Desarrollo.

1. A partir de la tabla de verdad dada, obtenga lo siguiente:

P	***Q***	***R***	***S***
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

a. La lista de maxterms
b. La lista de minterms
c. La función expresada como suma de productos
d. La función expresada como producto de sumas

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Introducción a la Teoría Computacional

Prueba #10

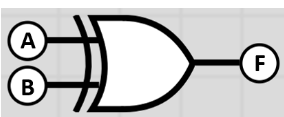
Nombre: ______________________ Cédula: ________________ Grupo: _______

Profesor: Dr. Carlos A. Rovetto   Puntos Obtenidos: _____ /

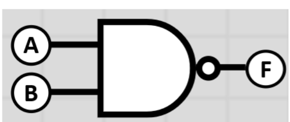
I Parte. Llene los espacios.

1. Gráficos que simbolizan una función lógica y son los bloques de cualquier circuito digital: _______________________________.

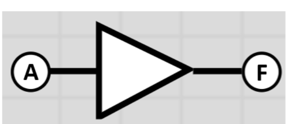
2. Indique el nombre de la función que representa cada compuerta lógica mostrada:



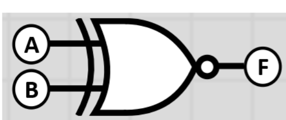
____



____



____



____

II Parte. Desarrollo.

1. Simplifique la siguiente función utilizando el método de mapa de Karnaugh.

$$F = A'B'C + A'BC' + A'BC + ABC$$

Anexos 2: Presentaciones

1

Capítulo I: Conceptos de Lógica y Razonamiento

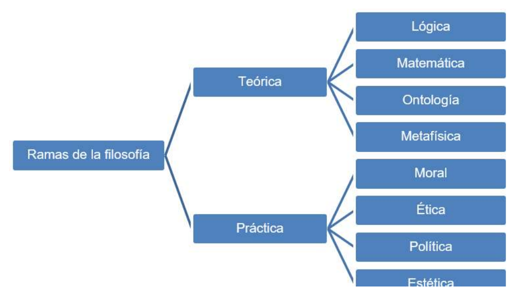
2

Capítulo I: Conceptos de lógica y razonamiento

En el Capítulo I de este documento se describirán conceptos básicos de lógica y razonamiento. Se presentará una introducción al estudio de la lógica como ciencia formal, brindando una definición a este concepto y una descripción acerca de su evolución a lo largo del tiempo. Se mencionarán otras disciplinas en las que se aplica la lógica, de modo que se pueda evidenciar la importancia del estudio de esta disciplina.

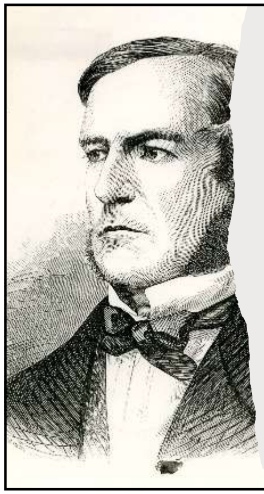
3

Definición y conceptos de lógica



La lógica es una de las ramas teóricas de la filosofía, cuyo vocablo de origen griego significa "amor a la sabiduría" y se encarga de explicar y entender la naturaleza, las causas o los principios de la realidad, del conocimiento o de los valores, a través del razonamiento.

4



La lógica

La lógica se puede definir como la disciplina que estudia el razonamiento a través de reglas y técnicas, que hacen posible determinar si un enunciado es cierto o falso.

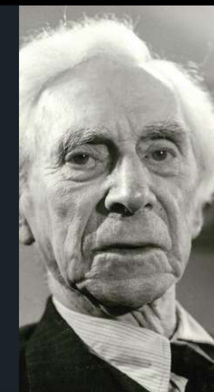
El origen de la lógica se fundamenta en los estudios realizados por *Aristóteles*, los cuales posteriormente los griegos utilizaron para demostrar las leyes matemáticas de manera formal.

Boole y *Augusto De Morgan*, quienes realizaron importantes aportes a la lógica matemática.

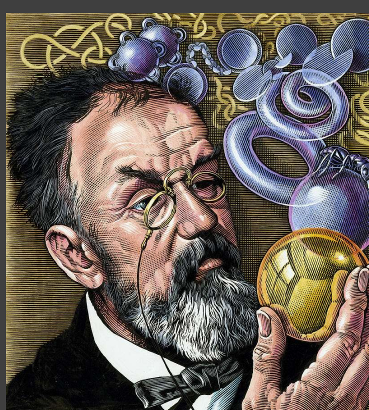
5

También destacan los trabajos de la Teoría de Conjuntos, realizados por los matemáticos alemanes *Gottlob Frege* y *Georg Cantor.*

Entre finales del siglo XIX e inicios del siglo XX, los matemáticos ingleses *Bertrand Russell* y *Alfred Whitehead* en sus estudios afirmaron que todas las leyes matemáticas pueden ser representadas a través proposiciones lógicas verdaderas, siendo esta teoría aún aceptada en la actualidad.



6



Parte del estudio de la lógica es establecer la validez de un argumento. Como define (Epp, 2011), un argumento es una secuencia de proposiciones (premisas) cuyo fin es demostrar que una afirmación (conclusión) es verdadera. Si las premisas de un argumento son verdaderas, entonces la conclusión también es verdadera y por consiguiente el argumento es válido.

- **Ejemplo #1:** Si todos los gatos son carnívoros y Coco es un gato, por lo tanto Coco es carnívoro.

Forma lógica: Si *p* y *q*, por lo tanto *r*.

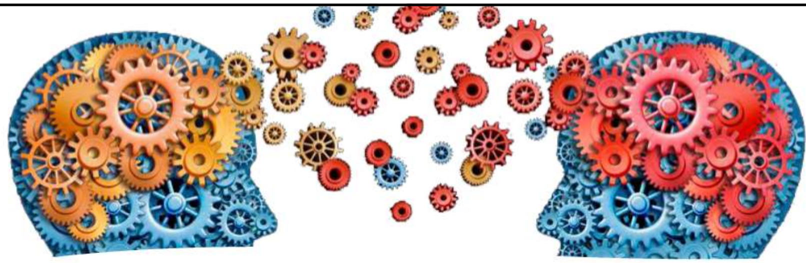
7

Tipos de lógica

Existen diversos tipos de lógica, como se muestra en la Figura Ambos sistemas formales, tanto la lógica de predicados como el lenguaje proposicional, consisten en las siguientes partes:

- **Sintaxis:** Alfabeto y reglas para la formación de fórmulas bien formuladas (*wffs*).
- **Semántica:** Preguntas que conectan modelos: cómo traducir un *wff* por ejemplo en lenguaje natural (interpretación) y cuáles son las condiciones exactas de su verdad (determinación de valores de verdad).
- **Prueba (proof):** Es el principio del cálculo, los axiomas y de las reglas de inferencia. Los teoremas se deducen de los axiomas por medio de reglas de inferencia. Los axiomas son verdades o premisas evidentes que no requieren una demostración. Así, deben ser lógicamente cierto.

8



Lógica proposicional

La lógica proposicional es una lógica en el nivel de la sentencia, en el que la unidad más pequeña es una oración. En este tipo de lógica no interesan las oraciones individuales o el significado de estas, sino saber si las oraciones son verdaderas o falsas.

También interesa saber si el hecho de que una oración sea verdadera o falsa depende de un conjunto de oraciones, por lo que de ser afirmativo, entonces es relevante saber cómo se obtiene. Estas oraciones se denominan proposiciones.

9

Lógica de Predicados

La lógica proposicional no es lo suficientemente poderosa como para representar todo tipo de aserciones que se utilizan en la informática y las matemáticas, o para expresar ciertos tipos de relación entre proposiciones, tales como la equivalencia.

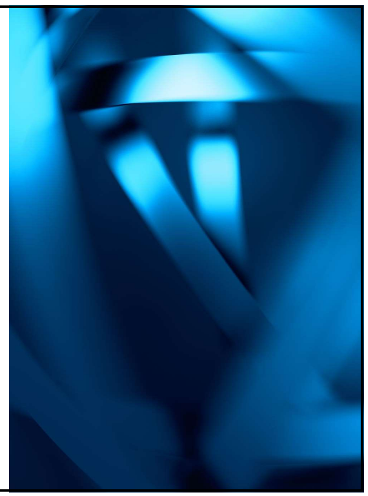
Por ejemplo, considere la siguiente aserción: “x es mayor que 1”. En esta aserción la *x* es una variable y por lo tanto, no es una proposición porque no se puede decir si es verdadera o falsa a menos que se conozca el valor de *x*.

10

Las **constantes** se utilizan para denotar nombres de criaturas del mundo real o ideal. De forma más precisa, la interpretación relaciona constantes a algunos elementos de un conjunto A. Las **variables** también se refieren a los elementos de un conjunto, pero no los determina explícitamente.

Un **predicado** es una plantilla de frase verbal que describe una propiedad de objetos o una relación entre objetos representados por las variables.

Los **cuantificadores** son frases especiales que a menudo se usan para cuantificar objetos en el discurso



11

Se pueden distinguir dos tipos de cuantificadores

- Cuantificador universal: Su símbolo es ∀ y se denota a través de la expresión ∀x P(x), la cual denota la cuantificación universal de la fórmula atómica P(x). Se lee “para todo *x*, *P(x)* tiene”, “para cada *x*, *P(x)* tiene” o “para cada *x*, *P(x)* mantiene”. En dicha expresión, la *x* significa *todos los objetos x en el universo* y como está seguido por *P(x)* entonces significa que *P(x)* es verdadero para cada objeto *x* en el universo.
- Cuantificador existencial: Su símbolo es ∃ y se denota a través de la expresión ∃x P(x), la cual denota la cuantificación existencial de la fórmula atómica P(x). Se lee “existe una x tal que *P(x)*” o “hay al menos un *x* tal que *P(x)*”. En dicha expresión, la *x* significa *al menos un objeto x en el universo* y como está seguido por *P(x)* entonces significa que *P(x)* es verdadero para al menos un objeto *x* en el universo.

12

Reglas para construir fórmulas bien formuladas (wff)

No todas las cadenas pueden representar proposiciones de la lógica de predicado. Para que los símbolos produzcan una proposición y sean bien interpretados deben seguir las reglas dadas abajo, y se llaman **wffs** (fórmulas bien formadas) de la lógica del predicado del primer orden. Un nombre de predicado seguido por una lista de variables tales como *P(x, y)*, donde P es un nombre de predicado y *x, y* son variables, se denomina fórmula atómica. Las *wffs* se construyen utilizando las siguientes reglas:

- Las *wffs* son verdadera y falsa.
- Cada constante proposicional (es decir, proposición específica), y cada variable proposicional (es decir, una variable que representa proposiciones) son *wffs*.
- Cada fórmula atómica (es decir, un predicado específico con variables) es un *wffs*.
- Si A, B y C son *wffs*, también lo son ¬ A, (A ∧ B), (A ∨ B), (A ⟶ B) y (A ⟷ B).

13

Teoría de Conjuntos

La teoría de conjuntos fue propuesta inicialmente por el matemático alemán Georg Cantor durante el final del siglo XIX, tras descubrir la potencial utilidad que ofrecía el estudio de los conjuntos en general en lugar de estudiar las propiedades de los elementos que componen a esos conjuntos.

Un conjunto es una colección de objetos considerada como un todo. Esta colección *está bien definida*, lo cual significa que para cada objeto que pueda formar parte de un conjunto, existe una forma de determinar si ese objeto forma o no parte del conjunto.

- Un conjunto se puede especificar de dos maneras:
- Por extensión: Enumerando sus elementos.
- Por comprensión: Indicando alguna propiedad que verifique todos y cada uno de los elementos del conjunto, y sólo a ellos.

14

También existen algunos conjuntos especiales, los cuales se definen a continuación:

- Conjunto universal: También denominado universo, es aquel que está formado por todos los elementos que se están considerando. Se representa por U, A o Ω. Como explica (Lipschutz & Lipson, 2009), todos los conjuntos en cualquier aplicación de la teoría de conjuntos pertenecen a un gran conjunto universal.
- Conjunto vacío: Es aquel que no contiene elementos y se denota por el símbolo $\emptyset$ o como {}. El conjunto vacío también se considera
- Conjunto unitario: Es aquel que está formado por un único elemento.

15

Operaciones

Existen varias formas de obtener nuevos conjuntos a partir de otros existentes. Las operaciones básicas son las siguientes:

- **Unión:** Si se tienen dos conjuntos A y B, la unión es el conjunto de todos los elementos de A o de B. Se denota por $A \cup B$. Simbólicamente, se representa como $A \cup B = \{x \mid x \in A \vee x \in B\}$.
- **Intersección:** Si se tienen dos conjuntos A y B, la intersección es el conjunto de los elementos que pertenecen tanto a A como a B. Se denota por $A \cap B$. Simbólicamente, se representa como $A \cap B = \{x \mid x \in A \wedge x \in B\}$.
- **Diferencia:** Si se tienen dos conjuntos A y B, la diferencia de A menos B es el conjunto de los elementos que pertenecen a A pero no pertenecen a B. Se denota por $A - B$. Simbólicamente, se representa como $A - B = \{x \mid x \in A \wedge x \notin B\}$.
- **Complemento:** Si se tiene un conjunto A, el complemento de A es el conjunto de todos los elementos que pertenecen al universo, pero que no pertenecen a A. Se denota por A^C, A' o $\bar{A}$. Simbólicamente, se representa como $A^C = \{x \mid x \in U \wedge x \notin A\}$.

16

Productos cartesianos y relaciones

Una relación se define como la correspondencia por medio de ciertas características, entre dos elementos de dos conjuntos. Es decir, si se tienen dos conjuntos A y B, el primer elemento a del conjunto A está relacionado con el segundo elemento b del conjunto B. Se denota como aRb.

Los tipos de relaciones básicas de conjuntos que se pueden identificar son los siguientes, descritos por (Conradie & Goranko, 2015):

- Relación reflexiva: Todo elemento de un conjunto A está relacionado consigo mismo, de modo que se cumple que $(a, a) \in R$.
- Relación simétrica: Si se tienen dos conjuntos A y B, se tiene una relación simétrica si se cumple que $(a, b) \in R$ y $(b, a) \in R$. También puede cumplirse que $(a, b) \notin R$ y $(b, a) \notin R$.
- Relación transitiva: Si se tienen tres conjuntos A, B y C, se tiene una relación transitiva si se cumple que $(a, b) \in R$ y $(b, c) \in R$ por lo que $(a, c) \in R$.

17

Diagramas de Venn

Los diagramas de Venn fueron desarrollados por el matemático y filósofo inglés John Venn (1834 - 1923).

Un diagrama de Venn es un gráfico que representa la relación entre los elementos de un conjunto, con regiones encerradas dentro de un plano (Lipschutz & Lipson, 2009). El universo se representa con un rectángulo y los conjuntos con regiones circulares. Se sombrea el área que se desea ilustrar.

18

Capítulo II: Sistema axiomático y su aplicación

19

Capítulo II: Sistema axiomático y su aplicación

En el Capítulo II de este documento se estudiarán el concepto de sistema axiomático y sus aplicaciones. Se definirá el concepto de sistema axiomático y se describirán sus elementos y sus propiedades. También se definirán los conceptos de equivalencia, tautología y contradicción, los cuales incluyen leyes que son muy útiles al momento de hacer demostraciones basadas en sistemas axiomáticos.

20

Definiciones básicas y símbolos

Un sistema axiomático, también denominado sistema formal o cálculo axiomático, es una lista de conceptos y hechos básicos a partir de los cuales se derivan otros conceptos y teoremas, a través de la definición y la deducción (Contreras Oré, 2017). Todo sistema axiomático se compone de los siguientes elementos (Garrido, 2005):

a) Un lenguaje que se compone de una tabla de símbolos (alfabeto) y de reglas para formar fórmulas (gramática).

b) Una lista de axiomas que son las fórmulas del sistema.

c) Un conjunto de reglas de inferencia que permite generar nuevas fórmulas a partir de una o más fórmulas dadas. A una nueva fórmula generada se le denomina conclusión.

21

Los axiomas de un sistema axiomático pueden tener las siguientes propiedades, basado en lo descrito por (Contreras Oré, 2017):

Los axiomas son consistentes si no hay contradicciones entre ellos; es decir, si se deduce una proposición, a la vez no se puede deducir su negación.

- Un axioma es independiente si no se puede deducir a partir de los otros axiomas.
- Los axiomas son completos si cada afirmación puede ser probada por axiomas.

Los axiomas de un sistema axiomático pueden estar expresados de manera formal o informal:

Axiomatización formal: Usa un lenguaje formal y cada axioma es una cadena finita de signos en el alfabeto del lenguaje formal, siguientes reglas combinatorias que hacen de la secuencia una fórmula bien formada.

Axiomatización informal: Usa una lengua natural formalizada y definiciones no ambiguas.

22

- Los valores de verdad de una proposición se pueden representar a través de una tabla de verdad. A continuación, de acuerdo con (Epp, 2011), se describen los valores de verdad de los operadores lógicos más comunes:
- Negación: Si p es una proposición, su negación es "no p", lo cual se representa con los símbolos $\neg$ o $\sim$. El valor de verdad de $\neg p$ es el opuesto al de p, es decir, si p es verdadero, $\neg p$ es falso y si p es falso, $\neg p$ es verdadero.
- Conjunción: Si p y q son proposiciones, la conjunción de ambas es "p y q", lo cual se representa con el símbolo $\wedge$. El valor de verdad será verdadero si y sólo si tanto p y q son verdaderas. De otro modo, si p o q son falsas o ambas son falsas, entonces $p \wedge q$ es falso.
- Disyunción: Si p y q son proposiciones, la disyunción de ambas es "p o q", lo cual se representa con el símbolo $\vee$. El valor de verdad de $p \vee q$ será verdadero si p es verdadero, si q es verdadero o si tanto p como q son verdaderas y será falso si tanto p como q son falsos

23

Equivalencia, Tautología y Contradicción

Dos proposiciones p y q son lógicamente equivalentes si tienen los mismos valores en su tabla de verdad (Jiménez Murillo, 2009). Esto es, si p es verdadero cuando q también es verdadero. Esto se representa como $p \equiv q$ o $p \Leftrightarrow q$. El uso de una tabla de verdad es muy útil cuando se quiere verificar si dos proposiciones son lógicamente equivalentes.

De igual manera, también se puede utilizar una serie de proposiciones de equivalencias lógicas que son útiles al momento de demostrar teoremas. A continuación, basado en (Lipschutz & Lipson, 2009), se listan algunas de estas proposiciones:

- Leyes de idempotencia
- Leyes asociativas
- Leyes conmutativas
- Leyes distributivas
- Doble negación
- Leyes de De Morgan

24

La Tautología

Una tautología es una proposición compuesta, representada con una letra mayúscula, que es verdadera para todos los valores de verdad de sus variables (Jiménez Murillo, 2009). Una tautología muy común es la de $p \vee \neg p$.

Todas las tautologías listadas a continuación tienen la forma $P \Rightarrow Q$, lo cual se lee como "si P entonces Q".

- Adición: $p \Rightarrow (p \vee q)$
- Simplificación: $(p \wedge q) \Rightarrow p$
- Absurdo: $(p \rightarrow q) \Rightarrow \neg p$
- Modus ponens: $p \wedge (p \rightarrow q) \Rightarrow q$
- Modus tollens: $(p \rightarrow q) \wedge \neg q \Rightarrow \neg p$

25

Las reglas de inferencia más utilizadas son las siguientes:

Modus ponens (PP): Proviene del latín "modus ponendo ponens". Esta regla de inferencia establece que si una primera afirmación implica una segunda afirmación y la primera afirmación es verdadera, entonces la segunda afirmación es verdadera.

Modus tollens (TT): Proviene del latín "modus tollendo tollens". Esta regla de inferencia establece que si una primera afirmación implica una segunda afirmación y la segunda afirmación no es verdadera, entonces la primera afirmación no es verdadera.

26

Condicionales y Bicondicionales

Una proposición condicional es aquella en la que se conectan dos enunciados y la verdad del segundo enunciado q, está condicionada por la verdad del primer enunciado p. Como explica (Lipschutz & Lipson, 2009), las proposiciones condicionales se utilizan mucho, principalmente en las matemáticas.

- Donde se pueden distinguir dos partes, como describe (Epp, 2011):
- La hipótesis o antecedente, que es la parte antes del operador, es decir p.
- La conclusión o consecuente, que es la parte después del operador, es decir q.

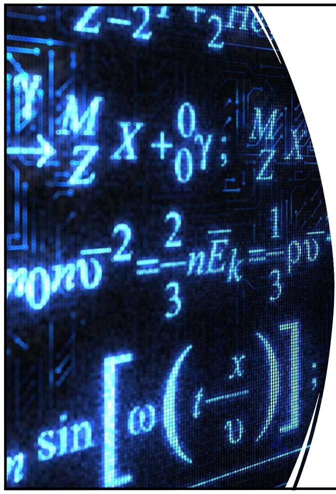
Una proposición bicondicional conecta dos enunciados p y q, donde p es verdadero si y sólo si q es verdadero. De igual manera, p es falso si y sólo si q es falso.

27

Existen algunas variaciones de la proposición condicional, las cuales se definen a continuación basado en las descripciones de (Epp, 2011) y (Jenkyns & Stephenson, 2018):

- **Inverso:** Para una proposición condicional $p \rightarrow q$, el inverso es $\neg p \rightarrow \neg q$. Es decir, es la negación tanto del antecedente como del consecuente de la proposición original.
- **Recíproco:** Para una proposición condicional $p \rightarrow q$, el recíproco es $q \rightarrow p$. Es decir, el consecuente pasa a ser el antecedente y el antecedente pasa a ser el consecuente.
- **Contrapositivo:** Para un proposición $p \rightarrow q$, el contrapositivo es $\neg q \rightarrow \neg p$. Es decir, es la negación del antecedente y del consecuente del inverso de la proposición original.

28



Cuantificadores

Un cuantificador convierte una declaración simbólica acerca de cualquier elemento del universo en una declaración acerca del universo (Stein, Drysdale, & Bogart, 2011). Es decir, especifican *cuántos* elementos del universo tienen una determinada propiedad, en lugar de indicar cuáles objetos tienen dicha propiedad. En este sentido, los cuantificadores se refieren a la cantidad de elementos para los cuales un predicado dado es verdadero.

- Cuantificador universal: Afirma que una declaración acerca de una variable es verdadera para todos los valores de la variable en el universo.
- Cuantificador existencial: Afirma que cierto elemento del universo existe. El símbolo utilizado para denotar este cuantificador es ∃ y se lee “existe” o “hay algunos”.

29

Cálculo proposicional

La lógica proposicional se encarga de estudiar cómo las proposiciones se relacionan unas con otras y si cada proposición es una declaración verdadera o falsa (O’Regan, 2013) (Levin, 2018). En este sentido, el cálculo proposicional, también denominado *lógica de orden 0*, estudia el álgebra de las proposiciones, por lo que no admite el uso de predicados y por lo tanto, no se usan variables ni cuantificadores.

30

Sistema Axiomático

Recordemos que un sistema axiomático se compone de un lenguaje, de axiomas y de reglas de inferencia, y que a partir de los axiomas se derivan otros razonamientos y proposiciones. Utilizando los axiomas a través de una demostración se pueden obtener teoremas.

31

Capítulo III: Elementos fundamentales de la teoría de conmutación

32

Capítulo III: Elementos fundamentales de la teoría de conmutación

En el Capítulo III de este documento se estudiarán conceptos y elementos fundamentales de la teoría de conmutación. Esto incluye el estudio del álgebra booleana, sus elementos, propiedades y teoremas los cuales serán muy útiles para la resolución de problemas. El álgebra booleana tiene numerosas aplicaciones en las ciencias computacionales y por lo tanto, es de gran importancia comprender este tema. Adicionalmente, se describirá más detalladamente el concepto de función lógica y se estudiarán las diversas formas en las que se puede representar una función lógica, mostrando.

33

Álgebra Booleana

El álgebra booleana, también llamada álgebra de Boole, en informática y matemática, es una estructura algebraica que esquematiza las operaciones lógicas.

El álgebra booleana fue desarrollada por George Boole y en su libro *"An Investigation of the Laws of Thought",* publicado en 1854, describió las herramientas para lograr que proposiciones lógicas puedan manipularse de manera algebraica (Jiménez Murillo, 2009). Sin embargo, no fue hasta 1938 cuando esto empezó a tener una aplicación directa, gracias a que la compañía estadounidense de teléfonos *"Bell"*, utilizó estas herramientas para analizar los circuitos de su red telefónica.

34

Dentro de la lógica booleana existen una serie de definiciones básicas:

- Variable: Símbolo que puede ser sustituido por un elemento del conjunto $B = \{0, 1\}$.
- Constante: Es un valor que pertenece al conjunto $\{0, 1\}$.
- Expresión: Se compone de variables, constantes y operadores.
- Función: Una función booleana de n variables $f(x_1, x_2, \dots, x_n)$, es una expresión que mapea f a un valor del conjunto booleano B.
- Literal: Es una variable o su complemento.

35

Teoremas del álgebra de Boole

El álgebra de Boole se rige por propiedades y reglas denominadas leyes o postulados. Estas permiten simplificar expresiones lógicas o transformarlas en expresiones equivalentes.

Todas las expresiones booleanas se caracterizan por su dualidad. El *dual* de una expresión booleana se basa en la expresión original reemplazando las operaciones AND por operaciones OR y viceversa, y reemplazando las constantes lógicas 0 por 1 y viceversa

36

Representación de funciones lógicas

Una función lógica, también denominada función booleana, describe las operaciones necesarias para que determinadas entradas de un circuito produzcan una salida que sólo puede tomar los valores de 0 y 1

El valor de una función lógica es equivalente al de una expresión de álgebra booleana y dependerá de los valores asignados a las variables, así como de las operaciones que componen a la función. Como explica (LaMeres, 2017), cada función lógica describe una sola salida de un circuito; sin embargo, si el circuito tiene múltiples salidas se necesitará una función lógica para cada salida.

- Producto lógico $(\cdot)$: Un producto lógico tendría la forma $F = A \cdot B$, donde el valor de F será 0 o 1 según los valores de las variables A y B.
- Suma lógica $(+)$: Una suma lógica tendría la forma $F = A + B$, donde el valor de F será 0 o 1 según los valores de las variables A y B
- Negación $(-)$: La negación invierte el valor de una variable, por lo que si una variable tiene el valor de 0 su negación tomará el valor de 1 y viceversa

37

Tabla de Verdad

Una tabla de verdad es una especificación formal que describe el comportamiento exacto de un circuito para todas las posibles entradas, especificando las salidas para cada combinación de entradas (Kumar Sarkar, Kumar De, & Sarkar, 2014). En una tabla de verdad se despliegan los valores de todas las combinaciones posibles de las variables y el valor que se le asocia a la función lógica.

38

Formas canónicas

Una forma canónica es un término estándar con el cual se representa una función y permite comparar funciones booleanas que están expresadas en términos algebraicos

- Suma de productos (SOP): También denominada forma canónica disyuntiva, es aquella expresión en la que una o más productos lógicos de uno o más literales, están conectados por operadores OR.
- Producto de sumas (POS): También denominada forma canónica conjuntiva, es aquella expresión en la que una o más sumas lógicas de uno o más literales, están conectados por operadores AND.

39

Conversión de una Forma a otras

Las funciones lógicas pueden representarse de diversas maneras y debido a esto, es posible pasar de una representación a otra. A continuación, se presentan diferentes ejemplos en donde se describirá cómo convertir de:

- Función a tabla de verdad
- Función en forma canónica a tabla de verdad
- Tabla de verdad a función en forma canónica

40

Funciones básicas

Generalmente, las funciones lógicas son muy complejas. Sin embargo, estas siempre se componen de una combinación de las funciones lógicas básicas, que son las mismas operaciones básicas del álgebra booleana.

- NOT (— o ′): En esta función representada como $F = \bar{A}$, el valor de la salida es la negación del valor de la entrada, es decir se invierte el valor de la variable. Dicho de otro modo, si la entrada tiene un valor de 0, la salida tendrá un valor de 1; mientras que si la entrada tiene un valor de 1, la salida tendrá un valor de 0.
- AND (·): En esta función representada como $F = A \cdot B$ o como $F = AB$, el valor de la salida será igual a 0 si al menos una de las entradas tiene el valor de 0; de lo contrario, si todas las entradas tienen el valor de 1, entonces el valor de la salida será 1.
- OR (+): En esta función representada como $F = A + B$, el valor de la salida será igual a 1 si al menos una de las entradas tiene el valor de 1; de lo contrario, si todas las entradas tienen el valor de 0, entonces el valor de la salida será 0.

41

A partir de las funciones anteriores se derivan el resto de las funciones lógicas, ya que se forman con una combinación de ellas. Estas se describen a continuación:

- XOR: También denominada OR exclusiva, en esta función representada como $F = A \oplus B$ o $F = \bar{A}B + A\bar{B}$, el valor de la salida será igual a 1 si alguna de las dos entradas tiene un valor de 1; de lo contrario, cuando las dos entradas tienen un valor de 0 al mismo tiempo, entonces la salida será igual a 0
- NAND (|): Esta función resulta de invertir la salida de una función AND. Se representa como $F = \overline{AB}$ o $F = \bar{A} + \bar{B}$ y el valor de la salida será igual a 0 si tanto la entrada A como la entrada B tienen un valor de 1; de lo contrario, la salida será igual a 0.
- NOR (↓): Esta función resulta de invertir la salida de una función OR. Se representa como $F = \overline{A + B}$ o $F = \bar{A} \cdot \bar{B}$ y el valor de la salida será igual a 1 si tanto la entrada A como la entrada B tienen un valor de 0; de lo contrario, la salida será igual a 0.
- XNOR: También denominada NOR exclusiva, esta función resulta de invertir la salida de una función NOR. Se representa como $F = \overline{A \oplus B}$ o $F = AB + \bar{A} \cdot \bar{B}$ y el valor de la salida será igual a 1 si ambas entradas tienen un valor de 0 al mismo tiempo o si ambas tienen un valor de 0 al mismo tiempo.

42

Implementación mediante conjuntos completos

Un conjunto funcionalmente completo se refiere a aquel conjunto de funciones que son necesarias y suficientes, de modo que de forma general todas las funciones lógicas pueden expresarse con las funciones AND, OR y NOT (Vingron, 2012).

- Representación de la suma lógica con el conjunto {AND, NOT}
- Representación del producto lógico con el conjunto {OR, NOT}
- Representación de la suma lógica y el producto lógico con el conjunto
- {NAND}
- Representación de la suma lógica y el producto lógico con el conjunto
- {NOR}

43

Capítulo IV: Puertas y Funciones Lógicas

44

Capítulo IV: Puertas y funciones lógicas

En el Capítulo IV de este documento se definirá el concepto de compuerta lógica y se describirán los ocho tipos de compuertas básicas que existen, presentando sus correspondientes expresiones algebraicas, sus tablas de verdad y sus diagramas de temporización.

45

Puertas lógicas

Las compuertas lógicas son gráficos que simbolizan una función lógica (Vingron, 2012), y son los bloques básicos de cualquier circuito digital, ya que estos tienen la capacidad de tomar decisiones. Como sucede con las funciones lógicas, el comportamiento de una compuerta puede describirse mediante:

Tabla de verdad: Representa de forma ordenada todas las posibles combinaciones de estados lógicos que pueden existir en las entradas y el valor que toma la salida en cada caso.

Expresión algebraica: Relaciona matemáticamente la salida con las entradas.

Diagrama de temporización: Representa gráficamente el comportamiento de una compuerta con señales variables en el tiempo.

46

- **Compuerta YES o Buffer**

Se representa con la expresión algebraica $\boldsymbol{F = A}$ y su comportamiento es el siguiente:
  - Si su entrada es cero, su salida será cero.
  - Si su entrada es uno, su salida será uno.

- **Compuerta AND**

Se representa con la expresión algebraica $\boldsymbol{F = A \cdot B}$ o $\boldsymbol{F = AB}$, y su comportamiento es el siguiente:
  - Si todas sus entradas son uno, su salida será uno.
  - Si al menos una de sus entradas es cero, su salida será cero.

- **Compuerta OR**

Se representa con la expresión algebraica $\boldsymbol{F = A + B}$ y su comportamiento es el siguiente:
  - Si al menos una de las entradas es uno, su salida será uno.
  - Si todas las entradas son cero, su salida será cero.

- **Compuerta NOT**

Se representa con la expresión algebraica $\boldsymbol{F = A'}$ o $\boldsymbol{F = \overline{A}}$, y su comportamiento es el siguiente:
  - Si su entrada es cero, su salida será uno.
  - Si su entrada es uno, su salida será cero.

47

- **Compuerta NAND**

Se representa con la expresión algebraica $\boldsymbol{F = \overline{AB}}$ o $\boldsymbol{F = \overline{A} + \overline{B}}$, y su comportamiento es el siguiente:
  - Si al menos una de sus entradas es cero, su salida será uno.
  - Si todas sus entradas son uno, su salida será cero.

- **Compuerta NOR**

Se representa con la expresión algebraica $\boldsymbol{F = \overline{A + B}}$ o $\boldsymbol{F = \overline{A} \cdot \overline{B}}$, y su comportamiento es el siguiente:
  - Si sus entradas son cero, su salida será uno.
  - Si al menos de sus entradas son uno, su salida será cero.

- **Compuerta XOR**

Se representa con la expresión algebraica $\boldsymbol{F = A \oplus B}$ o $\boldsymbol{F = \overline{A}B + A\overline{B}}$, y su comportamiento es el siguiente:
  - Si el número de entradas con valor igual a uno es impar, su salida será uno. De lo contrario, su salida será cero.

- **Compuerta XNOR**

Se representa con la expresión algebraica $\boldsymbol{F = \overline{A \oplus B}}$ o $\boldsymbol{F = AB + \overline{A} \cdot \overline{B}}$, y su comportamiento es el siguiente:
  - Si el número de entradas con valor igual a uno o cero es par, su salida será uno. De lo contrario, su salida será cero.

48

Simplificación de funciones lógicas

Como describe (Jiménez Murillo, 2009), muchas veces la función lógica obtenida a partir de un determinado problema no es óptima, por lo que es conveniente simplificar la expresión para facilitar la implementación de la función dada utilizando compuertas lógicas.

Para la simplificación de funciones lógicas se utilizan los teoremas y propiedades del álgebra booleana, de modo que se pueda obtener una función equivalente a la función que se está simplificando. A continuación, en la Tabla 28, se presenta un resumen de las propiedades y teoremas del álgebra booleana.

Teoremas			
1a	$A + 0 = A$	1b	$A \cdot 1 = A$
2a	$A + 1 = A$	2b	$A \cdot 0 = 0$
3a	$A + A = A$	3b	$A \cdot A = A$
4a	$A + A' = 1$	4b	$A \cdot A' = 0$
5a	$A + B = B + A$	5b	$A \cdot B = B \cdot A$
6a	$A + (B + C) = (A + B) + C$	6b	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
7a	$(A + B)' = A' \cdot B'$	7b	$(A \cdot B)' = A' + B'$
8a	$A + (BC) = (A + B)(A + C)$	8b	$A(B + C) = (AB) + (AC)$
9a	$A + (A \cdot B) = A$	9b	$A \cdot (A + B) = A$
10a	$(A + B)(A + B') = A$	10b	$(A \cdot B) + (A \cdot B') = A$
11a	$A + (A' \cdot B) = A + B$	11b	$A \cdot (A' + B) = AB$
12a	$(A')' = A$		

49

Forma canónica de una función lógica

	Suma de productos (SOP)	Producto de sumas (POS)
Descripción	Función en la que productos lógicos están conectados por operadores OR.	Función en la que sumas lógicas están conectados por operadores AND.
Cómo obtenerla a partir de una tabla de verdad	Tomar las filas en las que la función es igual a 1 y las variables en esas filas cuyo valor es igual a 0, se colocan complementadas en la expresión SOP.	Tomar las filas en las que la función es igual a 0 y las variables en esas filas cuyo valor es igual a 1, se colocan complementadas en la expresión POS.

Anteriormente, se estudiaron las formas canónicas en las que puede estar una función lógica: la suma de productos (SOP) y el producto de sumas (POS). Un resumen de las descripciones de cada una de las forma canónicas, se presenta a continuación en la Tabla

50

Maxterms y Mixterms

En el caso de una suma de productos, el término estándar recibe el nombre de minterm; mientras que en el caso de un producto de sumas, el término estándar recibe el nombre de maxterm.

- Minterm

Un minterm se define como un término producto en el que cada variable del problema aparece exactamente una vez, complementada o no (Katz & Borriello, 2005). Se denota de la forma m_i, donde i corresponde a la fila de la tabla de verdad en la que la función es exactamente igual a 1.

- Maxterm

Un maxterm se define como un término suma en el que cada variable del problema aparece exactamente una vez, complementada o no (Katz & Borriello, 2005). Se denota de la forma M_i, donde i corresponde a la fila de la tabla de verdad en la que la función es exactamente igual a 0.

51

Simplificación de funciones

Anteriormente, se estudió la simplificación de funciones lógicas mediante el uso de los teoremas del álgebra booleana. Sin embargo, como describe (Marcovitz, 2010), existen algunas desventajas, las cuales se mencionan a continuación:

Algunas veces no se puede estar seguro de haber obtenido la mínima expresión de una función.

Es posible que no se encuentre la mínima expresión de una función, aunque parezca que no se puede hacer nada más.

El proceso de simplificación se complica cuando se tienen cuatro o más variables.

Se requiere de habilidad para manipular las funciones, ya que no siempre se aplican los mismos teoremas o propiedades y el orden de aplicación de estos varía según la función.

52

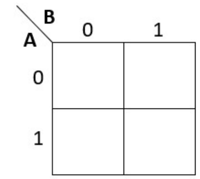
Diagramas de Karnaugh

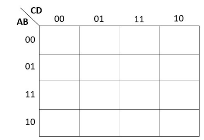
El método de diagrama de Karnaugh o mapa de Karnaugh fue propuesto inicialmente por Edward W. Veitch y luego modificado por Maurice Karnaugh (Jiménez Murillo, 2009). Un mapa de Karnaugh es un diagrama en forma de tabla, en el que cada celda contiene un minterm de la función. Como explica (Marcovitz, 2010), el mapa de Karnaugh permite encontrar de forma gráfica los términos más adecuados para simplificar una función expresada como suma de productos o como producto de sumas.

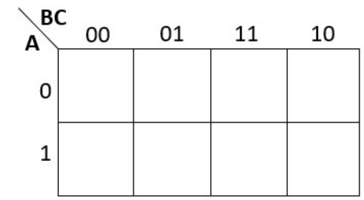
53

Diagramas de Karnaugh

- Para 2 variables: Se tiene que $n = 2$, por lo tanto $2^2 = 4\, celdas$. El mapa de Karnaugh quedaría de la siguiente manera
- Para 3 variables: Se tiene que $n = 3$, por lo tanto $2^3 = 8\, celdas$. El mapa de Karnaugh quedaría de la siguiente manera
- Para 4 variables: Se tiene que $n = 4$, por lo tanto $2^4 = 16\, celdas$







54